IT Fundamentals

Hoofdstuk 1: Getallensystemen of talstelsels

Inhoud

Inhoud

Inhoud

1.1. Inleiding

1.1.1. Voorstelling van getallen

- Getallen kunnen op verschillende manieren voorgesteld worden:

17

XVII

10001

1.1. Inleiding

1.1.2. Definities

- Een cijfer is een symbool, dat gebruikt wordt bij de voorstelling van getallen.
- Een getal wordt voorgesteld door, al dan niet van elkaar verschillende, cijfers achter elkaar te plaatsen
- Voorbeeld:
  - Wij gebruiken **10 symbolen** namelijk

    0, 1, 2, 3, 4, 5, 6, 7, 8, 9 -> onze cijfers
  - Door de cijfers achter elkaar te plaatsen krijgen we een getal:

    1213

1.1. Inleiding

1.1.3 Positie van een cijfer

- **Voorbeeld:**

1213

duizendtal honderdtal tiental eenheid

- Afhankelijk van de positie in het getal heeft het cijfer dus een andere betekenis.
- De positie van een cijfer binnen het getal geeft aan met hoeveel het cijfer vermenigvuldigd moet worden

$$1213 = 1 \,.\, 1000 + 2 \,.\, 100 + 1 \,.\, 10 + 3 \,.\, 1$$

$$= 1 \,.\, 10^3 + 2 \,.\, 10^2 + 1 \,.\, 10^1 + 3 \,.\, 10^0$$

1.2. Positionele systemen

- **Definitie:**

Een positioneel (getallen)systeem of positioneel talstelsel (kortweg positiestelsel), is een talstelsel, waarbij een getal wordt voorgesteld door een reeks van symbolen of cijfers. De positie of de plaats van het cijfer in het getal, bepaalt de bijdrage aan het getal op basis van een gekozen grondtal.

- **Algemeen:** een positioneel systeem heeft een
  - **Grondtal a**
  - **Verzameling symbolen of cijfers**, waarvan het aantal gelijk is aan a

1.2. Positionele systemen

1.2.1 Tiendelige getallen

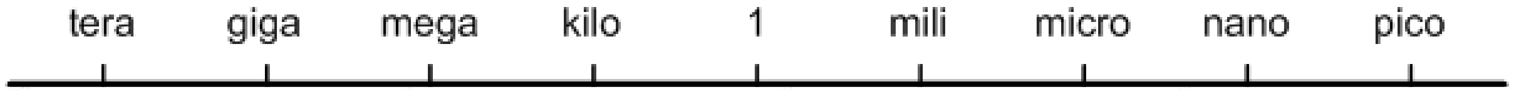
- Een positioneel systeem
- Decimaal stelsel
- Positioneel systeem met
  - Grondtal = **10**
  - Verzameling cijfers = {**0, 1, 2, 3, 4, 5, 6, 7, 8, 9**}

Bijvoorbeeld het getal 1302:

	10^3	10^2	10^1	10^0
X	1000	100	10	1
	1	3	0	2
+	1000	300	0	2
	1000 + 300 + 2 = 1302			

1.2. Positionele systemen

1.2.1 Tiendelige getallen - Decimale veelvouden



benaming	symbool	macht van 10	aantal eenheden
tera	T	10^{12}	1000 000 000 000
giga	G	10^{9}	1 000 000 000
mega	M	10^{6}	1 000 000
kilo	k	10^{3}	1 000
milli	m	10^{-3}	0,001
micro	µ	10^{-6}	0,000 001
nano	n	10^{-9}	0,000 000 001
pico	p	10^{-12}	0,000 000 000 001

1.2. Positionele systemen

1.2.2 Binaire getallen - Voorstelling

- Grondtal = **2**
- Verzameling symbolen = {0, 1}
- Voorbeeld getal 10100010110
- Betekenis van elk symbool:

	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	1024	512	256	128	64	32	16	8	4	2	1
X	1	0	1	0	0	0	1	0	1	1	0
	1024	0	256	0	0	0	16	0	4	2	0
+	1024 + 256 + 16 + 4 + 2 = 1302										

1.2. Positionele systemen

1.2.2 Binaire getallen - Toepassingsgebied

- Voorstelling informatie binnenin een computer

  Voorbeeld: een IPv4-adres bestaat uit 32 bits

  Probleem: lang → onoverzichtelijk → kansen op fouten

  Oplossing:

  binaire getal voorstellen in een ander talstelsel → dotted decimal

11000000 10101000 00001010 00000001

192 . 168 . 10 . 1

1.2. Positionele systemen

1.2.2. Binaire getallen - Afspraken en notatie

- Onderscheid tussen talstelsels:

  haakjes rond het getal met grondtal als subscript

  - $(10)_{10}$ = 10 als tiendelig getal
  - $(10)_{2}$ = 10 als binair getal

- **Afspraak:** indien geen notatie van grondtal, dan nemen we aan dat het getal in het decimaal stelsel staat

1.2. Positionele systemen

1.2.2. Binaire getallen - Binair tellen

Methode (analoog als bij decimaal tellen):

1. Begin bij 0
2. Vervang het laatste cijfer door zijn opvolger
3. Wanneer een 1 (hoogste cijfer) verhoogd moet worden, verhoog dan het cijfer van één rang hoger met één
4. Ga verder met stap 2

Binaire waarden	Decimale waarden
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13

1.2. Positionele systemen

1.2.2. Binaire getallen - Definities

- **Definities:**
  - **bit** =
    - binary digit
    - afgekort **b**
    - 1 cijfer (0 of 1) uit binair getal
  - **byte** = rij van 8 bits
    - afgekort **B**
- **msb** = **m**ost **s**ignificant **b**it / **lsb** = **l**east **s**ignificant **b**it

Voorbeeld getal: 10100010110

msb lsb

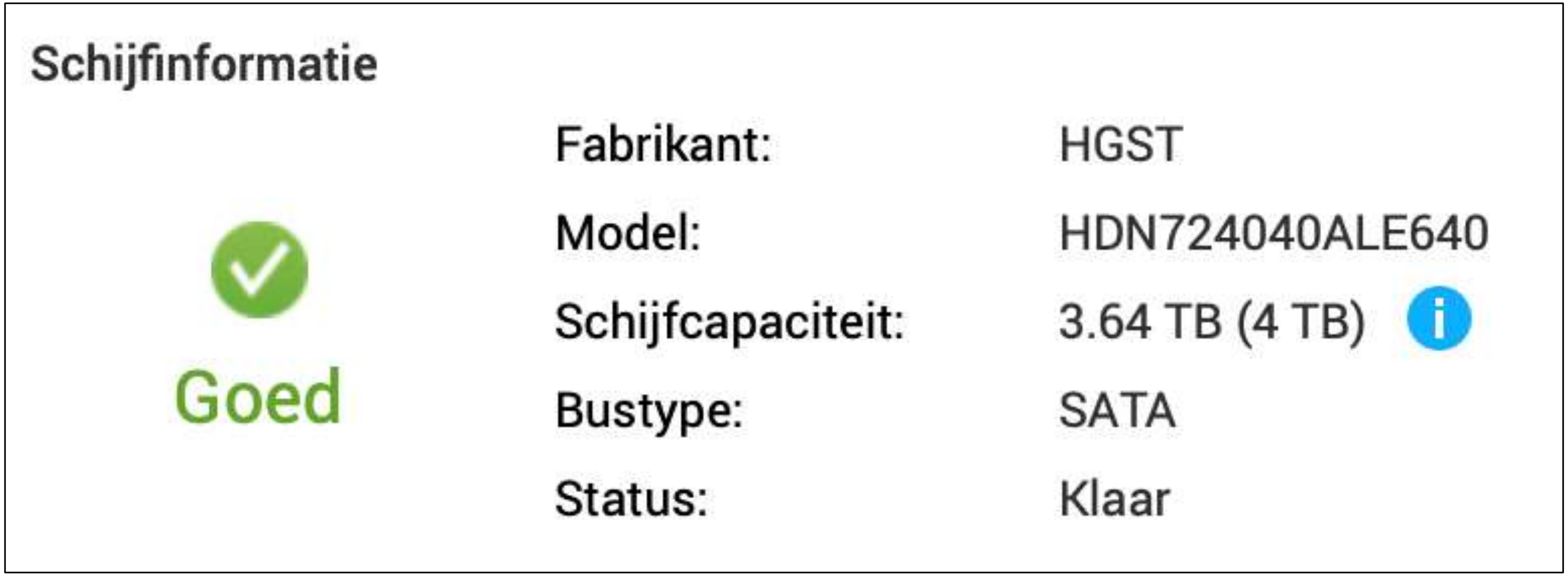
1.2. Positionele systemen

1.2.2. Binaire getallen - Veelvouden van bytes

SI-voorvoegsels		Binaire voorvoegsels		
Symbool (naam)	Waarde	Symbool (naam)	Waarde	Afwijking tussen SI en binair
kB (kilobyte)	$1000^1 = 10^3$	KiB (kibibyte)	$1024^1 = 2^{10}$	2,4%
MB (megabyte)	$1000^2 = 10^6$	MiB (mebibyte)	$1024^2 = 2^{20}$	4,9%
GB (gigabyte)	$1000^3 = 10^9$	GiB (gibibyte)	$1024^3 = 2^{30}$	7,4%
TB (terabyte)	$1000^4 = 10^{12}$	TiB (tebibyte)	$1024^4 = 2^{40}$	10,0%
PB (petabyte)	$1000^5 = 10^{15}$	PiB (pebibyte)	$1024^5 = 2^{50}$	12,6%
EB (exabyte)	$1000^6 = 10^{18}$	EiB (exbibyte)	$1024^6 = 2^{60}$	15,3%
ZB (zettabyte)	$1000^7 = 10^{21}$	ZiB (zebibyte)	$1024^7 = 2^{70}$	18,1%
YB (yottabyte)	$1000^8 = 10^{24}$	YiB (yobibyte)	$1024^8 = 2^{80}$	20,9%

1.2. Positionele systemen

1.2.2 Binaire getallen - Veelvouden van bytes (vervolg)



De waarde tussen haakjes is de schijfcapaciteit in decimale notatie (1GB = 1.000.000.000 bytes). De meeste schijffabrikanten gebruiken de decimale notatie. Het NAS toont de opslagruimte in binaire notatie (1GB = 1.073.741.824 bytes), die lager kan lijken maar dezelfde hoeveelheid ruimte is.

1.2. Positionele systemen

1.2.3 Octale getallen - Voorstelling

- Grondtal = **8**
- Verzameling symbolen = {**0, 1, 2, 3, 4, 5, 6, 7**}
- Voorbeeld getal $(172)_8$
- Betekenis van elk symbool:

8^2	8^1	8^0
1	7	2

- **Opmerking:**
  - Om onderscheid te maken met de tiendelige getallen worden octale getallen in IT-omgevingen vooraf gegaan door de prefix **0** (nul).

    Voorbeeld: $(172)_8$ = 0172

1.2. Positionele systemen

1.2.3. Octale getallen - Octaal tellen

- **Methode** analoog als bij decimaal en binair tellen
- Binair getal omzetten naar octale voorstelling:
    - binair getal groeperen per 3 bit (→ $2^3 = 8$)
      Eventueel leidende nullen toevoegen
      (groen in de tabel)
    - octale waarde bepalen per 3 bit

Voorbeeld: $(111010)_2 = (\ 111\ 010\ )_2 = (72)_8$

Octale waarden	Binaire waarden	Decimale waarden
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7
10	001 000	8
11	001 001	9
12	001 010	10
13	001 011	11
14	001 100	12
15	001 101	13

1.2. Positionele systemen

1.2.3. Octale getallen - Toepassingsvoorbeeld

- Bestandspermissies in Linux:
  - Bestaat uit 3 octale cijfers:
    - **Algemeen:**

      - $(rwx)_{owner}$ $(rwx)_{group}$ $(rwx)_{all}$ *bestandsNaam*
    - Hierbij staat:
      - r-bit voor ‘read’ of leesrecht
      - w-bit voor ‘write’ of schrijfrecht
      - x-bit voor ‘execute’ of recht van uitvoeren
    - Afhankelijk van wat toegelaten is (= 1) of niet toegelaten is (= 0) kunnen de bestandsrechten weergegeven worden in bit-vlaggen. Bijvoorbeeld:

      $(111)_{owner}$ $(101)_{group}$ $(100)_{all}$ *bestandsNaam*
    - Octale weergave van de bestandspermissies, wordt dan:

      $(7)_{owner}$ $(5)_{group}$ $(4)_{all}$ *bestandsNaam* of 754 *bestandsNaam*

1.2. Positionele systemen

1.2.4. Hexadecimale getallen - Voorstelling

- Grondtal = **16**
- Verzameling symbolen = {**0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A** (=10), **B** (=11), **C** (=12), **D** (=13), **E** (=14), **F** (=15)}
- **Voorbeeld:** getal $(1272)_{16}$
- Betekenis van elk symbool:

16^3	16^2	16^1	16^0
1	2	7	2

- **Opmerking:**
  - Om onderscheid te maken met de tiendelige getallen worden hexadecimale getallen in IT-omgevingen vooraf gegaan door de prefix **0x** (nul). Dus $(1272)_{16}$ = 0x1272

1.2. Positionele systemen

1.2.4. Hexadecimale getallen - Hexadecimaal tellen

- **Methode** analoog als bij decimaal, binair en octaal tellen
- Binair getal omzetten naar hexadecimale voorstelling:
  - binair getal groeperen per 4 bit (→ $2^4 = 16$) Eventueel leidende nullen toevoegen (groen in de tabel)
  - hexadecimale waarde bepalen per 4 bit

Voorbeeld: $(111010)_2 = (0011\ 1010)_2 = (3A)_{16}$

Hexadecimale waarden	Binaire waarden	Decimale waarden
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15
10	0001 0000	16
11	0001 0001	17

1.2. Positionele systemen

1.2.4. Hexadecimale getallen -Toepassingsvoorbeelden

- **IPv6-adres**:
  - Opvolger van IPv4-adres
  - 128 bits
  - 32 hexadecimale getallen verdeeld in groepjes van 4
    Voorbeeld: $(2001:0DB8:ACAD:0001:0000:0000:0000:0002)_{16}$
- **MAC-adres:**
  - Fabrikantafhankelijk fysiek adres van een netwerkkaart
  - 48 bits
  - 12 hexadecimale getallen
  - Verschillende voorstellingen: gegroepeerd per 2 of per 4
    Voorbeelden:
    - 00:0C:6E:C1:22:4A
    - 20c1-9bbf-324e

1.3. Conversies tussen talstelsels

1.3.1. Binaire combinaties

- **grondtal 2 → 2 symbolen** namelijk **0** en **1**

  1 bit => 2 mogelijkheden: 0 en 1

  2 bits => 4 mogelijkheden: 00, 01, 10, 11

  …

  n bits => $\mathbf{2^n}$ mogelijkheden

1.3. Conversies tussen talstelsels

1.3.1. Conversie decimaal binair

- Hoe $\mathbf{2^n}$ uitrekenen?
  - **Van buiten kennen**:
    $2^0 = \mathbf{1}$; $2^1 = \mathbf{2}$; $2^2 = \mathbf{4}$; $2^3 = \mathbf{8}$; $2^4 = \mathbf{16}$; … ; $\mathbf{2^{10} = 1024 = 1}$ **kilo** ; …
  - **Vergeet volgende rekenregels niet!:** $a^{n+m} = a^n.a^m$ **en** $a^n/a^m = a^{n-m}$
- **Voorbeeld 1:** Hoeveel mogelijke combinaties kunnen we vormen uit een getal met **9 bits**?
  - **Antwoord:** $2^9 = 512$ mogelijke combinaties
- **Voorbeeld 2:** Uit hoeveel bit moet een binair getal bestaan om **2048 mogelijke combinaties** te kunnen vormen?
  - **Antwoord:** $2048 = 2^{11}$ => het binaire getal moet uit 11 bits bestaan

1.3. Conversies tussen talstelsels

1.3.1. Conversie decimaal binair - Cijfers voor de komma (methode 1)

- Essentieel: de machten van 2 kennen:

2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1024	512	256	128	64	32	16	8	4	2	1

- Omzetten van binair naar decimaal:
  - Voorbeeld: $(1010)_2$
    - Vertrek vanaf de lsb, en voeg toe in de tabel van rechts naar links

	2^3	2^2	2^1	2^0
	8	4	2	1
Vermenigvuldigen:	1	0	1	0
	8	0	2	0
Optellen:	8+0+2+0 = 10			

- Naarmate men hier vlotter in wordt, kan men de machten uitschrijven:

  Voorbeeld: $(1010)_2 = 1.2^3 + 0.2^2 + 1.2^1 + 0.2^0$
  $= 1.8 + 1.2 = (10)_{10}$

1.3. Conversies tussen talstelsels

1.3.1. Conversie decimaal binair - Cijfers voor de komma (methode 1) (vervolg)

- Omzetten van decimaal naar binair
  - = herkennen van machten
  - Voorbeeld: $(862)_{10}$
    - Stap 1: zoek naar de hoogste macht van 2 kleiner dan het getal:

      $2^9 = 512$
    - Stap 2: trek deze waarde van het getal af :

      $862 - 512 =$ **350**
    - Herhaal stap 1 en 2 op het resultaat van stap 2 tot wanneer je 0 uitkomt:

      $2^8 = 256$ -> $350 - 256 =$ **94** → $2^6 = 64$ -> $94 - 64 =$ **30** → $2^4 = 16$ -> $30 - 16 =$ **14** →
      $2^3 = 8$ -> $14 - 8 = 6$ → $2^2 = 4$ -> $6 - 4 = 2$ → $2^1 = 2$ -> $2 - 2 = 0$
    - Maak een tabel van opeenvolgende machten van 2, vertrekkende van de hoogste gebruikte macht, en schrijf een 1 onder de gebruikte machten en een 0 onder de niet gebruikte machten:

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	0	1	0	1	1	1	1	0

- Antwoord: $(862)_{10} = (1101011110)_2$

1.3. Conversies tussen talstelsels

1.3.1. Conversie decimaal binair - Cijfers na de komma (methode 1)

- Essentieel: de machten van 2 kennen:

2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
0,5	0,25	0,125	0,0625	0,03125

- Omzetten van binair naar decimaal:
  - Voorbeeld: $(0{,}011)_2$
    - Vertrek na de msb (dus msb wordt niet toegevoegd in de tabel), en voeg toe in de tabel van links naar rechts

	2^{-1}	2^{-2}	2^{-3}
	0,5	0,25	0,125
Vermenigvuldigen:	0	1	1
	0	0,25	0,125
Optellen:	0+0,25+ 0,125 = 0,375		

  - Naarmate men hier vlotter in wordt, kan men de machten onmiddellijk uitschrijven:

Voorbeeld: $(0{,}011)_2 = 0.2^{-1} + 1.2^{-2} + 1.2^{-3}$
$= 1 \,.\, 0{,}25 + 1 \,.\, 0{,}125 = (0{,}375)_{10}$

1.3. Conversies tussen talstelsels

1.3.1. Conversie decimaal binair - Cijfers na de komma (methode 1) (vervolg)

- Omzetten van decimaal naar binair
  - = herkennen van machten
  - Voorbeeld: $(0{,}875)_{10}$
    - Stap 1: zoek naar de hoogste negatieve macht van 2 kleiner dan het getal:

      $2^{-1} = 0{,}5$
    - Stap 2: trek deze waarde van het getal af :

      $0{,}875 – 0{,}5 = \mathbf{0{,}375}$
    - Herhaal stap 1 en 2 op het resultaat van stap 2 tot wanneer je 0 uitkomt:

      $2^{-2} = 0{,}25$ -> $0{,}375 – 0{,}25 = \mathbf{0{,}125}$ → $2^{-3} = 0{,}125$ -> $0{,}125 – 0{,}125 = \mathbf{0}$
    - Maak een tabel van dalende negatieve machten van 2, vertrekkende van 2^{-1}, en schrijf een 1 onder de gebruikte machten en een 0 onder de niet gebruikte machten:

2^{-1}	2^{-2}	2^{-3}
1	1	1

    - Antwoord: $(0{,}875)_{10} = (0{,}111)_2$

1.3. Conversies tussen talstelsels

1.3.1. Conversie decimaal binair – Cijfers na de komma (methode 2)

- Uitschrijven of herkennen van de machten is enkel praktisch voor omzettingen binair/decimaal met beperkt aantal cijfers na de komma.

- Indien er na de komma teveel cijfers zijn en/of je geen herkenning van een macht kunt toepassen, val je terug op volgende methodes:

  - Voor omzetting binair naar decimaal: methode van de opeenvolgende delingen
  - Voor omzetting decimaal naar binair: methode van de opeenvolgende vermenigvuldigingen

1.3. Conversies tussen talstelsels

1.3.1. Conversie decimaal binair – Cijfers na de komma (methode 2) (vervolg)

- Omzetting binair naar decimaal: methode van de opeenvolgende delingen
- Voorbeeld:
  - Stoppen aan de komma

$(0,101)_2 = (\ldots)_{10}$

$(0 + 1)/2 = 0,5$

$(0,5 + 0)/2 = 0,25$

$(0,25 + 1)/2 = (0,625)_{10}$

1.3. Conversies tussen talstelsels

1.3.1. Conversie decimaal binair – Cijfers na de komma (methode 2) (vervolg)

- Omzetting decimaal naar binair: methode van de opeenvolgende vermenigvuldigingen
- Voorbeeld:

$(0{,}375)_{10} = (\ldots)_2$

$0{,}375 \,.\, 2 = 0{,}75 \rightarrow 0$ (cijfer voor de komma)

$0{,}75 \,.\, 2 = 1{,}5 \rightarrow 1$

$0{,}5 \,.\, 2 = 1 \rightarrow 1$

$(0{,}375)_{10} = (0{,}011)_2$

- Per lijn:
  - Vermenigvuldig je met 2
  - Hou je het cijfer voor de komma opzij
  - De volgende lijn doe je enkel verder met het deel na de komma van de voorgaande lijn en herhaal je voorgaande stappen
  - Je stopt wanneer er geen cijfers meer na de komma zijn als gevolg van de vermenigvuldiging
  - Het eindresultaat is het decimaal gedeelte, te lezen van boven naar onder

1.3. Conversies tussen talstelsels

1.3.3. Conversies talstelsels met als basis een macht van 2

- In geval dat de basis van een talstelsel een macht van 2 is, zal één symbool van dat talstelsel voorgesteld kunnen worden door een vast aantal bits.
- Voorbeelden:
  - Octaal talstelsel: basis = 8 = 2^3
    → dus voorstelling elk symbool door 3 bits
  - Hexadecimaal stelsel: basis = 16 = 2^4
    → dus voorstelling elk symbool door 4 bits
- Omzetten talstelsel naar binair:
  - Stel elke symbool voor door de overeenkomstige groep bits (zie tabellen volgende pagina)
- Omzetten binair naar talstelsel:
  - In het binaire getal symbolen groeperen, startend bij de komma.
  - Eventueel voor- en achteraan nullen toevoegen indien je geen volledig groepje kan vormen.

Octale waarden	Binaire waarden	Decimale waarden
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7
10	001 000	8
11	001 001	9
12	001 010	10
13	001 011	11
14	001 100	12
15	001 101	13

Hexadecimale waarden	Binaire waarden	Decimale waarden
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15
10	0001 0000	16
11	0001 0001	17

1.3. Conversies tussen talstelsels

1.3.3. Conversies talstelsels met als basis een macht van 2 (vervolg)

Voorbeelden:

$(11010{,}0100001)_2 = (0001\ 1010{,}0100\ 0010)_2 = (1A{,}42)_{\mathbf{16}}$

$(11010{,}0100001)_2 = (011\ 010{,}010\ 000\ 100)_2 = (32{,}204)_{\mathbf{8}}$

$(7{,}A8)_{\mathbf{16}} = (0111{,}1010\ 1000)_2 = (111{,}10101)_2$

$(4{,}17)_{\mathbf{16}} = (0100{,}0001\ 0111)_2 = (100{,}00010111)_2$

$(4{,}17)_{\mathbf{8}} = (100{,}001\ 111)_2 = (100{,}001111)_2$

1.3. Conversies tussen talstelsels

1.3.4. Oefeningen op conversies

Oefening 1:

Reken om

1. $(100110010011)_2 = (\ldots\ldots\ldots\ldots\ldots)_{16} = (\ldots\ldots\ldots\ldots\ldots)_8$
2. $(1001110011110010)_2 = (\ldots\ldots\ldots\ldots\ldots)_{16} = (\ldots\ldots\ldots\ldots\ldots)_8$
3. $(101010111100)_2 = (\ldots\ldots\ldots\ldots\ldots)_{16} = (\ldots\ldots\ldots\ldots\ldots)_8$
4. $(1FD)_{16} = (\ldots\ldots\ldots\ldots\ldots)_2$
5. $(CCC)_{16} = (\ldots\ldots\ldots\ldots\ldots)_2$
6. $(5307)_8 = (\ldots\ldots\ldots\ldots\ldots)_2$
7. $(7264)_8 = (\ldots\ldots\ldots\ldots\ldots)_2$
8. $(2A5C)_{16} = (\ldots\ldots\ldots\ldots\ldots)_8$
9. $(243)_8 = (\ldots\ldots\ldots\ldots\ldots)_{16}$

1.3. Conversies tussen talstelsels

1.3.4. Oefeningen op conversies (vervolg)

Oefening 2:

Reken om

1. $(100110{,}010011)_2 = (\ldots\ldots\ldots\ldots\ldots)_{16} = (\ldots\ldots\ldots\ldots\ldots)_8$
2. $(1011100{,}1111001)_2 = (\ldots\ldots\ldots\ldots\ldots)_{16} = (\ldots\ldots\ldots\ldots\ldots)_8$
3. $(10101{,}01111)_2 = (\ldots\ldots\ldots\ldots\ldots)_{16} = (\ldots\ldots\ldots\ldots\ldots)_8$
4. $(1F{,}D)_{16} = (\ldots\ldots\ldots\ldots\ldots)_2$
5. $(C{,}CC)_{16} = (\ldots\ldots\ldots\ldots\ldots)_2$
6. $(53{,}07)_8 = (\ldots\ldots\ldots\ldots\ldots)_2$
7. $(72{,}64)_8 = (\ldots\ldots\ldots\ldots\ldots)_2$
8. $(2A{,}5C)_{16} = (\ldots\ldots\ldots\ldots\ldots)_8$
9. $(2{,}43)_8 = (\ldots\ldots\ldots\ldots\ldots)_{16}$

1.3. Conversies tussen talstelsels

1.3.4. Oefeningen op conversies (vervolg)

Oefening 3:

Reken om

1. $(10000100)_2 = (\ldots\ldots\ldots\ldots\ldots)_{10}$
2. $(1000010001)_2 = (\ldots\ldots\ldots\ldots\ldots)_{10}$
3. $(100100001)_2 = (\ldots\ldots\ldots\ldots\ldots)_{10}$
4. $(131)_{10} = (\ldots\ldots\ldots\ldots\ldots)_2$
5. $(260)_{10} = (\ldots\ldots\ldots\ldots\ldots)_2$
6. $(15)_{10} = (\ldots\ldots\ldots\ldots\ldots)_2$

1.3. Conversies tussen talstelsels

1.3.4. Oefeningen op conversies (vervolg)

Oefening 4:

Reken om

1. $(1000{,}0110)_2$ = $(\ldots\ldots\ldots\ldots\ldots)_{10}$
2. $(111001)_2$ = $(\ldots\ldots\ldots\ldots\ldots)_{10}$
3. $(100010010{,}0011)_2$ = $(\ldots\ldots\ldots\ldots\ldots)_{10}$
4. $(10101100{,}011)_2$ = $(\ldots\ldots\ldots\ldots\ldots)_{10}$
5. $(0{,}1111)_2$ = $(\ldots\ldots\ldots\ldots\ldots)_{10}$
6. $(12{,}15)_8$ = $(\ldots\ldots\ldots\ldots\ldots)_{10}$
7. $(26{,}7)_8$ = $(\ldots\ldots\ldots\ldots\ldots)_{10}$
8. $(1A{,}4)_{16}$ = $(\ldots\ldots\ldots\ldots\ldots)_{10}$
9. $(1A{,}18)_{16}$ = $(\ldots\ldots\ldots\ldots\ldots)_{10}$

1.3. Conversies tussen talstelsels

1.3.4. Oefeningen op conversies (vervolg)

Oefening 5:

Reken om

1. $(11,5625)_{10} = (\ldots\ldots\ldots\ldots\ldots)_{2}$
2. $(36)_{10} = (\ldots\ldots\ldots\ldots\ldots)_{2}$
3. $(172)_{10} = (\ldots\ldots\ldots\ldots\ldots)_{2}$
4. $(17,375)_{10} = (\ldots\ldots\ldots\ldots\ldots)_{2}$
5. $(89,625)_{10} = (\ldots\ldots\ldots\ldots\ldots)_{8}$
6. $(126,25)_{10} = (\ldots\ldots\ldots\ldots\ldots)_{16}$
7. $(86,2)_{10} = (\ldots\ldots\ldots\ldots\ldots)_{16}$
8. $(67,3)_{10} = (\ldots\ldots\ldots\ldots\ldots)_{16}$
9. $(1984)_{10} = (\ldots\ldots\ldots\ldots\ldots)_{16}$

1.3. Conversies tussen talstelsels

1.3.4. Oefeningen op conversies (vervolg)

Oefening 6: IPv4-adres

- Conversies tussen het decimale, binaire en hexadecimale talstelsels worden voortdurend toegepast bij het berekenen van zowel IPv4- als IPv6-adressen.
- Een **IPv4-adres** bestaat uit **32 bits** opgedeeld in 2 delen namelijk een **netwerkwerkgedeelte** en een **hostgedeelte**.
- Het **netwerkgedeelte** vertelt je tot welk netwerk je apparaat behoort en het **hostgedeelte** vertelt je welk apparaat het binnen een netwerk is.
- Elk netwerk beschikt over een **netwerkadres**, waarbij alle bits van het hostgedeelte op **0** staan.
- Eveneens beschikt een netwerk over een **broadcastadres**, waarbij alle bits van het hostgedeelte op **1** staan.
- Elk netwerk beschikt ook over een **adresrange** waaruit een IP-adres kan gekozen worden om toe te kennen aan een apparaat binnen een bepaald netwerk. De adresrange is het interval tussen het netwerkadres en het broadcastadres van een netwerk.

1.3. Conversies tussen talstelsels

1.3.4. Oefeningen op conversies (vervolg)

Oefening 6: IPv4-adres (vervolg)

- Een IPv4-adres heeft twee voorstellingen namelijk
  - een **binaire** voorstelling en een
  - **dotted decimal** voorstelling.
- De **binaire** voorstelling bestaat uit **32 bits**.
- De **dotted decimal** voorstelling bestaat uit **4 decimale cijfers met een punt tussen**.
- Om over te stappen van de binaire voorstelling naar de dotted decimal voorstelling groeperen we de bitreeks per acht bit (= 1Byte), waarbij we van elk groepje de decimale waarde bepalen. Zo bekomen we bijvoorbeeld 192.168.16.5

1.3. Conversies tussen talstelsels

1.3.4. Oefeningen op conversies (vervolg)

Oefening 6: IPv4-adres (vervolg)

- **Gegeven:** Een IPv4-adres 192.168.78.64 waarbij de eerste 24 bit netwerkgedeelte zijn en de laatste 8 bit hostgedeelte.

- **Gevraagd:** Bereken
    1) het netwerkadres
    2) het broadcastadres
    3) de adresrange waartoe dit IP-adres behoort?

1.3. Conversies tussen talstelsels

1.3.4. Oefeningen op conversies (vervolg)

Oefening 6: IPv4-adres (vervolg)

- **Gegeven:** Een IPv4-adres 192.168.78.64 waarbij de eerste 24 bit netwerkgedeelte zijn en de laatste 8 bit hostgedeelte.
- **Oplossing:**

1.4. Bewerkingen in het binair stelsel

1.4.1. Optellen in het binair stelsel

Optellen in het decimaal stelsel:

- **Werkwijze:**
  - Schrijf de twee getallen onder elkaar **met de komma's onder elkaar**
  - Tel cijfer per cijfer bij elkaar op, van rechts naar links
  - Draag indien nodig een één (1) over naar een hogere rangorde

1	1				
1	5	7	,	3	7
2	7	9	,	4	
4	3	6	,	7	7

1.4. Bewerkingen in het binair stelsel

1.4.1. Optellen in het binair stelsel (vervolg)

Optellen in het binair stelsel:

- **Werkwijze (analoog aan decimaal stelsel):**
  - Schrijf de twee getallen onder elkaar **met de komma's onder elkaar**
  - Tel cijfer per cijfer bij elkaar op, van rechts naar links
  - Draag indien nodig een één (1) over naar een hogere rang

$$
\begin{array}{ccccccc}
1 & & & 1 & & & \\
 & 1 & 0 & 0 & , & 1 & 1 \\
 & 1 & 1 & 0 & , & 1 & \\
\hline
1 & 0 & 1 & 1 & , & 0 & 1
\end{array}
$$

1.4. Bewerkingen in het binair stelsel

1.4.2. Oefeningen op optellen in het binair stelsel

- Maak deze optellingen in het binair stelsel. Converteer waar nodig eerst naar binair om vervolgens de binaire optelling uit te voeren. Als controle kan je het resultaat opnieuw converteren naar het decimale talstelsel om te zien of je de optelling correct hebt uitgevoerd.

1. $(1011)_2 + (10101)_2 = (\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots)_2$
2. $(110{,}11)_2 + (10{,}101)_2 = (\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots)_2$
3. $(23{,}25)_{10} + (40{,}5)_{10} = (\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots)_2$
4. $(100)_{10} + (28)_{10} = (\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots)_2$
5. $(97)_{10} + (115)_{10} = (\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots)_2$
6. $(147)_{10} + (35)_{10} = (\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots)_2$

1.4. Bewerkingen in het binair stelsel

1.4.3. Andere bewerkingen

- **Complement(eren)**
  - **Alle enen worden nullen én omgekeerd**
  - **PAS OP**! alleen mogelijk wanneer er een beperkend kader is
    → Byte (8 bit); 16-, 32- of 64-bit woord
  - Operator: **¬**

    Voorbeeld: ¬[00011010] = [11100101]

- **Bitsgewijze EN**
  - Op twee operanden wordt **BIT PER BIT** de EN-bewerking toegepast (zie tabel)
  - Operator: **∧**

    Voorbeeld: [1010] **∧** [1100] = [1000]

∧	**0**	**1**
0	0	0
1	0	1

1.4. Bewerkingen in het binair stelsel

1.4.3. Andere bewerkingen (vervolg)

- **Bitsgewijze OF**
  - Operator : **V**

    Voorbeeld: [10001101] **V** [00111011] = [10111111]

V	**0**	**1**
0	0	1
1	1	1

- **Bitsgewijze Exclusieve OF of xor**
  - Operator : **<u>V</u> of xor**

    Voorbeeld: [10001101] **<u>V</u>** [00111011] = [10110110]

<u>V</u>	**0**	**1**
0	0	1
1	1	0

1.4. Bewerkingen in het binair stelsel

1.4.4. Negatieve getallen

- In het decimaal stelsel stellen we een negatief getal voor door een min-teken of '-' te plaatsen voor de absolute waarde. De bewerking het verschil of het aftrekken wordt gezien als de som van een positief en een negatief getal.

- Bij het binair stelsel zitten we met een bijkomend probleem omdat getallen per computerwoord (8, 16, 32 of 64 bit) bijgehouden worden in het geheugen en in de CPU.

  - De voorstelling van negatieve getallen zal dus binnen de beperkte ruimte van het computerwoord moeten gebeuren.
  - Het teken van een getal moet dus binnen het woord bijgehouden worden

1.4. Bewerkingen in het binair stelsel

1.4.4. Negatieve getallen

- 3 methodes om een negatief getal binair voor te stellen :
  1. **Teken + absolute waarde**
  2. **Excess-N**
  3. **Two’s complement of 2’s complement**

- We beperken ons in het vervolg tot 8-bit computerwoorden

1.4. Bewerkingen in het binair stelsel

1.4.4. Negatieve getallen - Teken + absolute waarde

- Meest linkse bit binnen het woord gebruiken als tekenbit
  - **0** → positief ; **1** → negatief
  - Voorbeelden: -12 → [10001100] en +22 → [00010110]

- **Nadelen:**
  1. A-B kan niet berekend worden als A+(-B): bijvoorbeeld 22-12 = 22 + (-12):

$$
\begin{array}{r}
00010110 \\
\underline{+\,10001100} \\
10100010
\end{array}
\quad = (\mathbf{-34})_{10}
$$

  2. Aftrekking moet als afzonderlijke bewerking in de hardware geïmplementeerd worden → duur!
  3. Bewerkingen kunnen pas uitgevoerd worden nadat beide tekenbits uit de woorden verwijderd en geanalyseerd werden.
  4. Nul heeft twee voorstellingen : [00000000] en [10000000]

1.4. Bewerkingen in het binair stelsel

1.4.4. Negatieve getallen - Excess-N

- N bij het getal optellen (**N = 127** bij 8 bit woorden)
  - Voorbeelden in een **8 bit computerwoord:**

$(1)_{10}$	→ 128	→ [10000000]
$(0)_{10}$	→ 127	→ [01111111]
$(-127)_{10}$	→ 0	→ [00000000]
$(-1)_{10}$	→ 126	→ [01111110]
$(128)_{10}$	→ 255	→ [11111111]

- **Voordeel:** Nul wordt slechts op één manier voorgesteld.
- **Nadelen:**
  - Rekenen met negatieve getallen wordt ook niet vereenvoudigd:
    Voorbeeld: $-127 + (-127) = -127$
  - Geen enkel getal wordt nog op de gewone binaire manier voorgesteld

1.4. Bewerkingen in het binair stelsel

1.4.4. Negatieve getallen - 2’s complement

- **Positieve getallen** → gewone binaire vorm

- **Negatieve getallen** → **2’s complement** van de binaire vorm
  - Werkwijze:
    - Schrijf de binaire vorm van de absolute waarde
    - **inverteer** hiervan alle bits (dus 0 → 1 en 1 → 0)
    - **tel** er **binair 1** bij **op**

- Voorbeelden:

$(0)_{10}$ → [00000000] $(1)_{10}$ → [00000001]

$(-1)_{10}$ → [11111111] $(-10)_{10}$ → [11110110]

  - **Opmerking:** een positief getal heeft bij de 2’s complement voorstelling een **0** als msb en een negatief getal heeft een **1** als msb

1.4. Bewerkingen in het binair stelsel

1.4.4. Negatieve getallen - 2's complement (vervolg)

- **Voordeel:** bij deze voorstelling geldt wél dat A-B = A+ (-B)

$(10)_{10}$ → [00001010]

$(-20)_{10}$ → [11101100]

[11110110]

- **Controle**:
  - [11110110] is een negatief getal, want msb = 1
  - Om de absolute waarde te bepalen:
    - Inverteer eerst de bits: [00001001]
    - Tel er binair 1 bij op: [00001010] = $(10)_{10}$

  → Dus oplossing = $(-10)_{10}$

1.4. Bewerkingen in het binair stelsel

1.4.5. Oefeningen op negatieve getallen

Oefening: Vul onderstaande tabel aan. Gebruik telkens een 8-bit computerwoord

getal	Teken + abs	Excess-127	2's-compl.
10			
-13			
43			
-52			
87			
-87			

1.4. Bewerkingen in het binair stelsel

1.4.5. Oefeningen op negatieve getallen (vervolg)

Oefening: Vul onderstaande tabel aan. Gebruik telkens een 8-bit computerwoord

getal	Teken + abs	Excess-127	2’s-compl.
106			
–106			
127			
–127			
128			
–128			

1.4. Bewerkingen in het binair stelsel

1.4.6. Overflow

- Sommige berekeningen, bij het optellen van twee 2's-complement getallen, lopen fout omdat het resultaat buiten het bereik van het computerwoord of register valt.
- Voorbeeld: -116 + (-60) = -176; binair wordt dit

$(-116)_{10} \rightarrow [10001100]$

$(-60)_{10} \rightarrow \underline{[11000100]}$

$1[01010000] \rightarrow (80)_{10}$

- We spreken van overflow omdat het computerwoord als het ware 'overloopt'.

1.4. Bewerkingen in het binair stelsel

1.4.6. Overflow

- Wanneer treedt overflow op?
  - Als er 1 van de **overflow condities** optreedt (niet beide samen!). Deze zijn:

1. **“carry naar het tekenbit”**
   = Overdracht van de op één na meest linkse positie naar de meest linkse positie (msb)
2. **“carry naar buiten”**
   = Overdracht van de meest linkse positie (msb) naar nergens of dus buiten het computerwoord

- Wanneer treedt **geen** overflow op?
  - Er een carry naar buiten **EN** een carry naar het tekenbit is
  - **Geen van beide** zich voordoen

Carry naar tekenbit	Carry naar buiten	Overflow	Correcte som
Nee	Nee	Nee	Ja
Nee	Ja	Ja	Nee
Ja	Nee	Ja	Nee
Ja	Ja	Nee	Ja

1.4. Bewerkingen in het binair stelsel

1.4.6. Overflow (vervolg)

Voorbeeld:

- $(100)_{10} + (50)_{10} = (150)_{10}$

		2's complement voorstelling	
		1 1	→ Carry naar tekenbit => overflow
$(100)_{10}$	=	[0 1 1 0 0 1 0 0]	
$(50)_{10}$	=	[0 0 1 1 0 0 1 0]	
		[1 0 0 1 0 1 1 0]	→ $(-106)_{10}$ => verkeerde som

1.4. Bewerkingen in het binair stelsel

1.4.6. Overflow (vervolg)

Voorbeeld:

- $(100)_{10} + (-50)_{10} = (50)_{10}$

	2's complement voorstelling	
	1 1 1 1	→ Carry naar tekenbit en carry naar buiten => GEEN overflow
$(100)_{10}$ =	[0 1 1 0 0 1 0 0]	
$(-50)_{10}$ =	[1 1 0 0 1 1 1 0]	
	[0 0 1 1 0 0 1 0]	→ $(50)_{10}$ => correcte som

1.4. Bewerkingen in het binair stelsel

1.4.6. Overflow (vervolg)

Voorbeeld:

- $(-100)_{10} + (-50)_{10} = (-150)_{10}$

	2's complement voorstelling	
	1 1 1 1	→ Carry naar buiten => overflow
$(-100)_{10}$ =	[1 0 0 1 1 1 0 0]	
$(-50)_{10}$ =	[1 1 0 0 1 1 1 0]	
	[0 1 1 0 1 0 1 0]	→ $(106)_{10}$ => verkeerde som

1.4. Bewerkingen in het binair stelsel

1.4.6. Overflow (vervolg)

Voorbeeld:

- $(-100)_{10} + (50)_{10} = (-50)_{10}$

	2's complement voorstelling	
	1 1	→ GEEN carry's => GEEN overflow
$(-100)_{10}$ =	[1 0 0 1 1 1 0 0]	
$(50)_{10}$ =	[0 0 1 1 0 0 1 0]	
	[1 1 0 0 1 1 1 0]	→ $(-50)_{10}$ => correcte som

1.4. Bewerkingen in het binair stelsel

1.4.7. Oefeningen op overflow

Oefening 1:

Maak deze berekeningen in 2's-complement voorstelling in een 8-bit woord en controleer het resultaat.

1. $(-22)_{10} - (12)_{10} \quad =$
2. $(11)_{10} - (5)_{10} \quad =$
3. $(-13)_{10} - (15)_{10} \quad =$
4. $(100)_{10} - (64)_{10} \quad =$

1.4. Bewerkingen in het binair stelsel

1.4.7. Oefeningen op overflow (vervolg)

Oefening 2:

- Maak deze berekeningen in binaire 2's-complement voorstelling en geef aan of en waarom er een overflow conditie is.

1. 55 - 25
2. 100 + 29
3. -125 + 25
4. -70 - 80
5. 43 + 106
6. 43 – 106
7. 87 – 52
8. -20 – 52
9. 87 – 127
10. 150 + 106

1.5. Floating-point

1.5.1. Hoe kommagetallen voorstellen

- In het **decimaal** stelsel kunnen we een kommagetal voorstellen op verschillende manieren:

$$0{,}005 = 0{,}05 \;.\; \mathbf{10}^{-1} = 0{,}5 \;.\; \mathbf{10}^{-2} = 5 \;.\; \mathbf{10}^{-3}$$

- In het binaire talstelsel kan je kommagetallen op dezelfde manier voorstellen als in het decimale talstelsel.

$$0{,}011 = 0{,}11 \;.\; \mathbf{2}^{-1} = 1{,}1 \;.\; \mathbf{2}^{-2} = 11 \;.\; \mathbf{2}^{-3}$$

1.5. Floating-point

1.5.2. Floating-point voorstelling

- Bij de voorstelling van een decimaal kommagetal is het aantal cijfers na de komma variabel. Vandaar dat we moeten bijhouden waar de komma staat.

- We hebben, bijgevolg, 2 getallen nodig om één kommagetal voor te stellen, namelijk
  - één met de waarde en
  - één met de aanduiding waar de komma moet komen (meestal onder de vorm van een exponent).

- Deze notatie noemen we de **wetenschappelijke notatie**.

Voorbeeld: 123456 = 1,23456 . 10^5 of 1,23456^{E}05

1.5. Floating-point

1.5.2. Floating-point voorstelling (vervolg)

- Ook bij de binaire voorstelling van een kommagetal is het aantal cijfers na de komma variabel.
- We hebben echter een bijkomend voordeel: in de binaire voorstelling van een kommagetal is het eerste beduidend cijfer of de msb steeds gelijk aan 1.
- Hiervan maken we gebruik om de genormaliseerde vorm te bepalen:
  - Bij de genormaliseerde vorm verschuiven we de komma zodat het eerste beduidende cijfer (=1) voor de komma staat.
- In deze genormaliseerde vorm hebben we dan nog
  - een getal nodig om de **mantisse** (of deel na de komma) te bepalen
  - En een getal om de **exponent** te bepalen.

Voorbeeld: $(101{,}1101)_2 = (1{,}\underbrace{011101}_{\text{mantisse}})_2 \cdot 2^{2}$ $\longrightarrow$ **exponent**

1.5. Floating-point

1.5.3. Floating-point in de computer

- Bij de voorstelling van floating-point getallen in de computer zullen we moeten afspreken hoeveel bits we gebruiken voor de mantisse en hoeveel voor de exponent.
- 4 mogelijk gestandaardiseerde formaten volgens de **IEEE 754**:

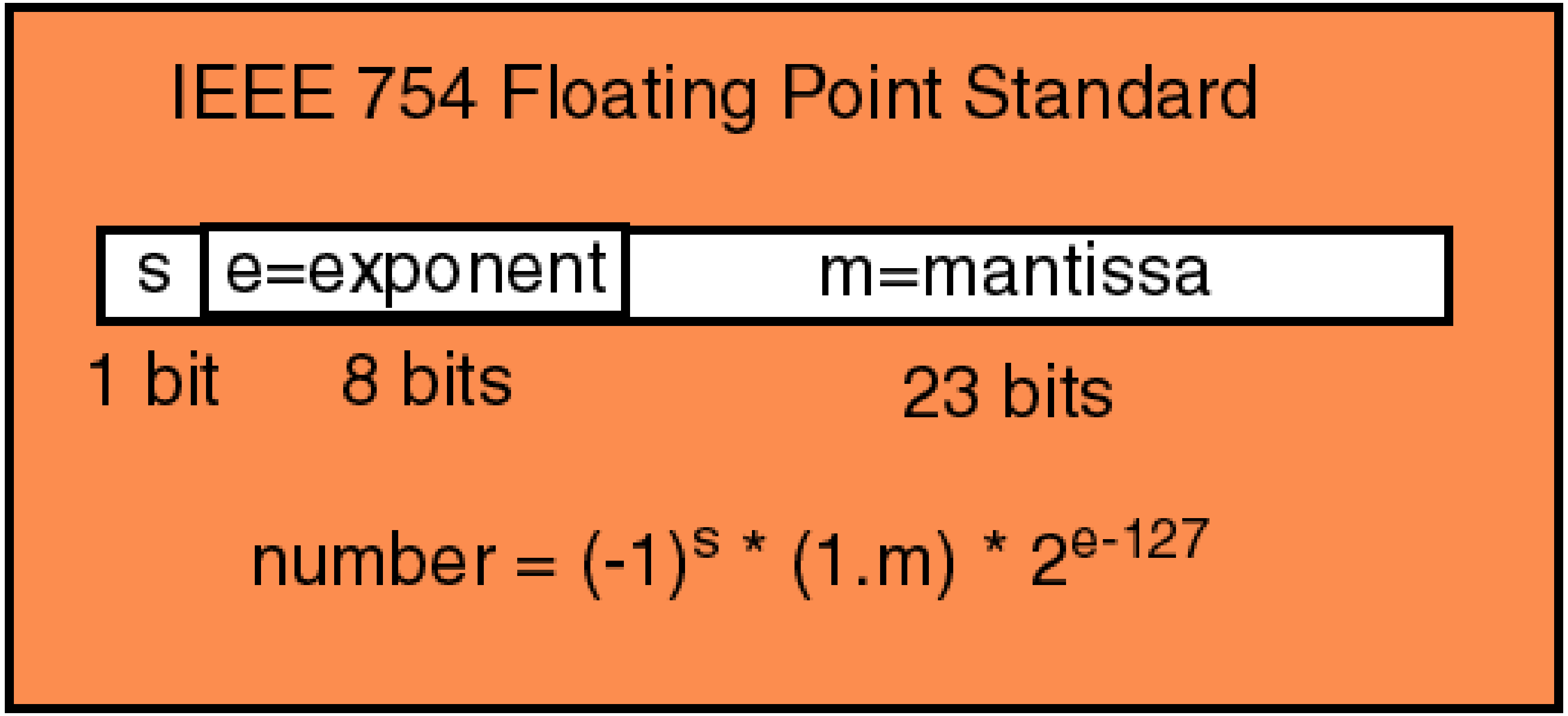
naam	precisie	sign-bit	exponent bits	mantisse bits	Excess -N	totaal
binary16	half-precision	1	5	10	15	16
binary32	single-precision	1	8	23	127	32
binary64	double-precision	1	11	52	1023	64
binary128	quadruple-precision	1	15	112	16383	128

- We zullen ons, bij de verdere bespreking, beperken tot het binary32-formaat of enkelvoudige precisie

1.5. Floating-point

1.5.3. Floating-point in de computer (vervolg)

- Een **IEEE 754** getal met enkelvoudige precisie heeft de volgende opbouw:



1.5. Floating-point

1.5.4. Decimale waarde van een IEEE 754 binary32 getal

- **Voorbeeld:**

  Er staat volgend binary32 getal in het geheugen:

  11000000101100000000000000000000

- **Welke waarde stelt dit voor?**

  - 1 bit voor **teken (s)**) → $(1)_2$ => het getal is negatief
  - 8 bits voor de **exponent (e)** voorgesteld in de excess-127 → $(10000001)_2 = (129)_{10}$ => 129-127 dus exponent 2
  - 23 bits voor de **mantisse (m)** → 011000…. → enkel het deel na de komma => 1,011000… = $(1{,}375)_{10}$

Het getal is dus $-1{,}375 \cdot 2^2 = (-5{,}5)_{10}$

1.5. Floating-point

1.5.5. IEEE 754 binary32 getalwaarde van een decimaal getal

- **Voorbeeld:** $(-\ 2{,}25)_{10}$

1ste stap: binaire voorstelling

=> $(-\ 10{,}01)_2$

2de stap: genormaliseerde vorm

=> $(-\ 1{,}001 \times 2^1)_2$

3de stap: exponent met Excess-127

=> $(127 + 1 = 128)_{10}$ => $(1000\ 0000)_2$

4de stap: IEEE 754 single voorstelling (32 bit)

Negatief getal ←

1	**1000 0000**	00100000000000000000000

1.5. Floating-point

1.5.6. Hexadecimale voorstelling van een IEEE 754 binary32 getal

- Omdat een binary32 getal nogal lang is, en bij overnemen er gemakkelijk fouten kunnen gemaakt worden, wordt er dikwijls gebruik gemaakt van de **hexadecimale voorstelling**.

  Bijvoorbeeld:

11000000101100000000000000000000

$\downarrow$ Verdelen in groepjes van 4

1100 0000 1011 0000 0000 0000 0000 0000

$\downarrow$ Omzetten naar hexadecimaal

$\mathbf{=(C0B00000)_{16}}$

1.5. Floating-point

1.5.7. Oefeningen op floating point

Oefening 1: bepaal de decimale waarde van het volgende binary32 getal in hexadecimale vorm: $(41E00000)_{16}$

teken	exponent								Mantisse																						

Teken = ……… => ………………………………

Exponent = (………………………………)$_2$

= (……………………)$_{10}$

=> …………………… = ……………………

Mantisse = (……………………………………………………………) $_2$

=> ……………………………………………

Het decimale getal is ………………………………………………

1.5. Floating-point

1.5.7. Oefeningen op floating point (vervolg)

Oefening 2:

- Bepaal voor volgende hexadecimale getallen de decimale waarde van hun floating-point voorstelling met **enkelvoudige precisie**:

1) $(CE100000)_{16}$ = ……………………………………

2) $(43C00000)_{16}$ = ……………………………………

3) $(3FC00000)_{16}$ = ……………………………………

4) $(42F48000)_{16}$ = ……………………………………

1.5. Floating-point

1.5.7. Oefeningen op floating point (vervolg)

Oefening 3:

Bepaal voor volgende decimale waarden hun floating-point voorstelling met **enkelvoudige precisie,** in hexadecimale vorm:

$(149{,}25)_{10}$ =

$(-1)_{10}$ =

$(538{,}625)_{10}$ =

$(-5/32)_{10}$ =

IT Fundamentals hoofdstuk 2

Logische Poorten

HO
GENT

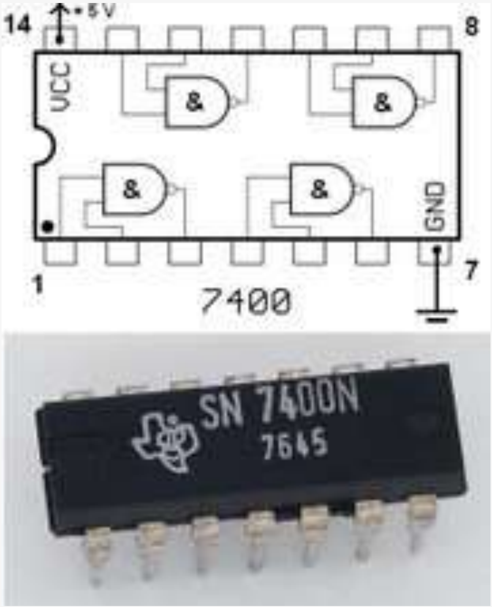
Inhoud.

HO GENT

2.1. Overzicht van een logische poort

Wat is een logische poort?

- Een **logische poort** is een digitale elektronische schakeling die werken volgens de (booleaanse) **logica**.
- Logische poorten zijn de elementaire bouwstenen van computers en andere digitale apparaten.
- Ze zijn opgebouwd uit elektronische componenten zoals transistors, weerstanden en dioden. Eén poort is opgebouwd uit o.a. 1 tot 10 transistors.
- Ze bevinden zich meestal in grote hoeveelheden samen op IC's (**I**ntegrated **C**ircuits), maar zijn ook als afzonderlijke bouwstenen beschikbaar.

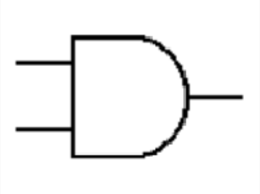


HO GENT

2.1. Overzicht van een logische poort

Eigenschappen van logische poorten

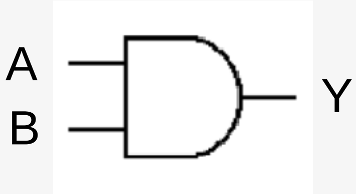
- De meest kenmerkende eigenschappen van logische poorten is:
  - Dat ze één of meerdere ingangen hebben en maar één uitgang
    - Zie figuur: logische EN-poort met 2 ingangen en 1 uitgang
  - De signalen (van de ingangen en uitgang) kunnen maar twee verschillende waarden aannemen, die kunnen geïnterpreteerd worden als 0 en 1 of Waar en Onwaar of Hoog en Laag. Men spreekt van:
    - Logische hoog of 1 bij een voltage van 3,5 volt of hoger
    - Logische laag of 0 bij een voltage van 0,5 volt en lager.



HO GENT

2.1. Overzicht van een logische poort

Waarheidstabel



- Wegens de verwantschap met de logica, kunnen we gebruik maken van waarheidstabellen om voor elke mogelijke combinatie van de ingangen de overeenkomstige waarde van de uitgang weer te geven
  - Dit laatste geldt zowel voor elke logische poort apart als voor een netwerk van logische poorten.
- Opbouw waarheidstabel
  - **Aantal kolommen** = aantal ingangen + aantal tussenwaarden + uitgang
    - In het voorbeeld 2 ingangen, nl. A en B en 1 uitgang, nl. Y
  - **Aantal rijen** = titelrij + $2^{\text{aantal ingangen}}$

    → rijen altijd in oplopende binaire volgorde
    - In het voorbeeld 2 ingangen, nl. A en B dus $1 + 2^2 = 5$ rijen

A	B	Y
0	0	…
0	1	…
1	0	…
1	1	…

2.2. Basispoorten

2.2.1. de NIET-poort

2.2.2. de EN-poort

2.2.3. de OF-poort

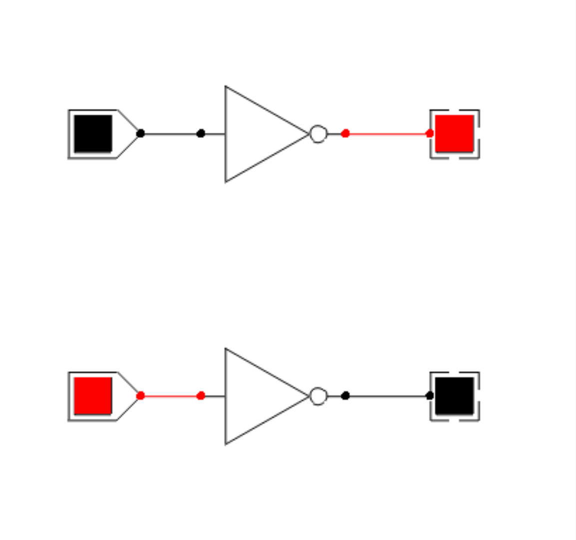
2.2.4. de XOR-poort

2.2.5. de NEN-poort

2.2.6. de NOR-poort

2.2. Basispoorten (vervolg)

- De logische **NIET** of **NOT** → komt overeen met complementeren in de logica



In	Uit
0	1
1	0

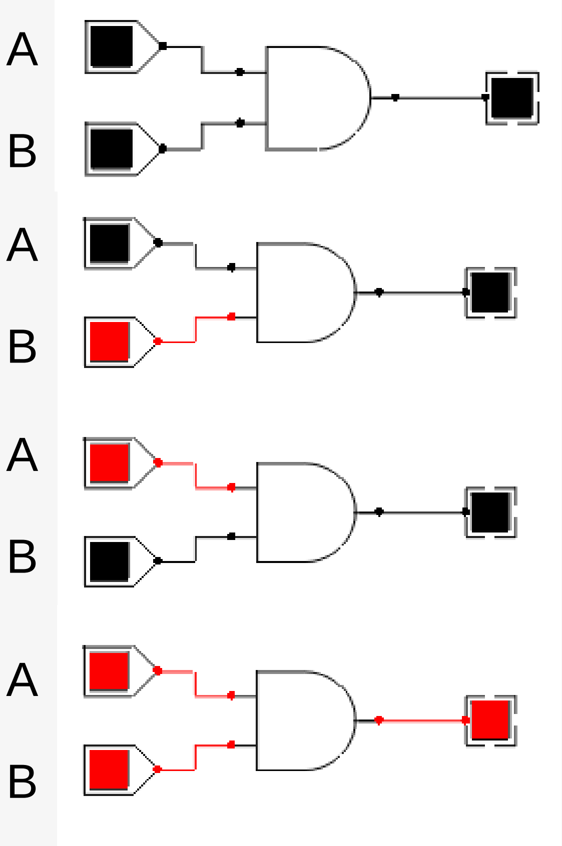
Opmerking:

- Signaal zwart → binair 0
- Signaal rood → binair 1

HO GENT

2.2. Basispoorten (vervolg)

- De logische **EN** of **AND** → komt overeen met **∧** in de logica

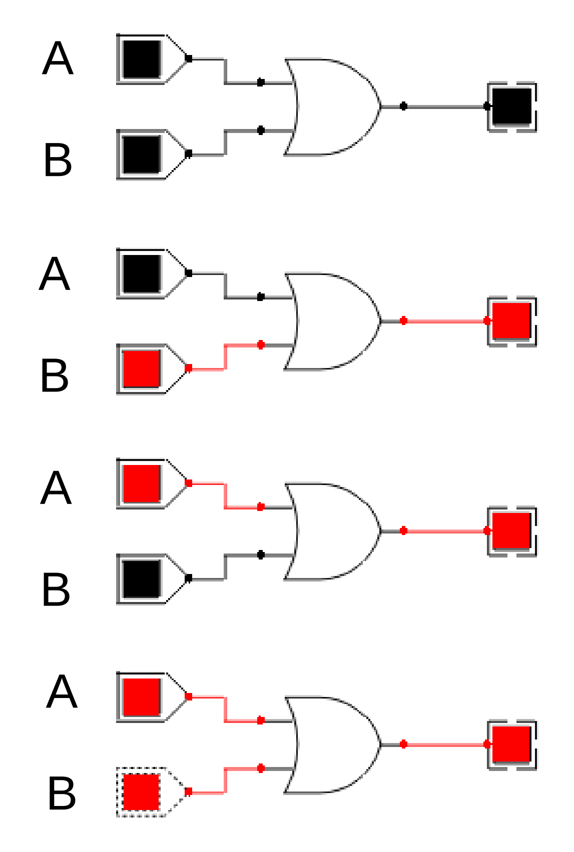


EN poort		
A	B	Uit
0	0	0
0	1	0
1	0	0
1	1	1

HO GENT

2.2. Basispoorten (vervolg)

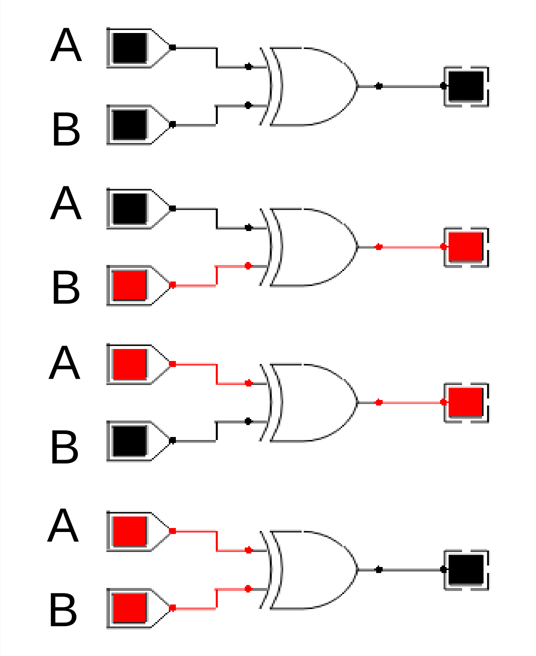
- De logische **OF** of **OR** → komt overeen met **V** in de logica



OF-poort		
A	B	Uit
0	0	0
0	1	1
1	0	1
1	1	1

2.2. Basispoorten (vervolg)

- De logische **Exclusieve OF** of **XOR** → wordt ook nog genoteerd als **V**



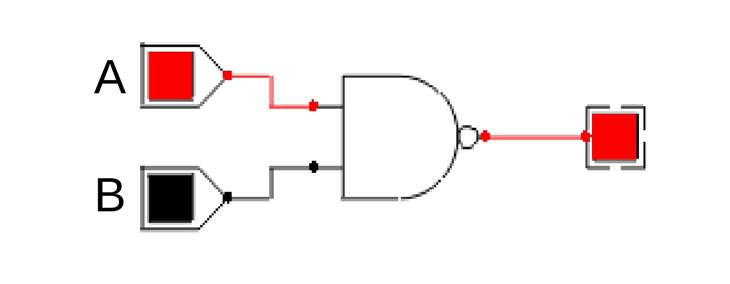
A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Functie → uitgang is 1 als de ingangen verschillend zijn en is 0 als de ingangen gelijk zijn.

HO GENT

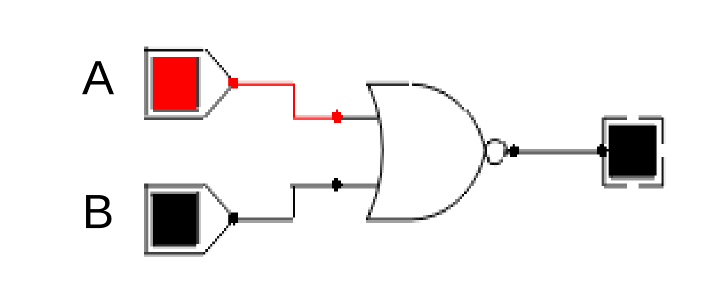
2.2. Basispoorten (vervolg)

- De logische **NEN** of **NAND** → komt overeen met **Λ** + complementeren in de logica



A	B	NEN
0	0	1
0	1	1
1	0	1
1	1	0

- De logische **NOF** of **NOR** → komt overeen met **V** + complementeren in de logica

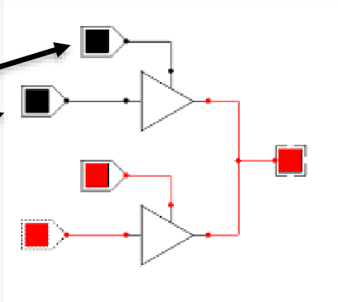


A	B	NOF
0	0	1
0	1	0
1	0	0
1	1	0

De tri-state buffer

- **Tri-state buffer**
  - Dit is een speciale poort die gebruikt wordt om kortsluiting (x/) op bussen te vermijden.

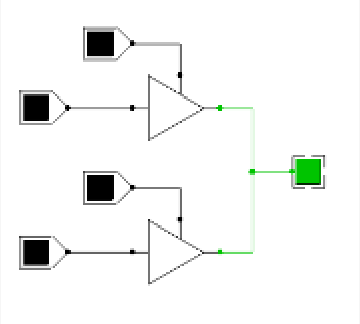
  - **Twee ingangen**:
    - stuurlijn
    - signaal



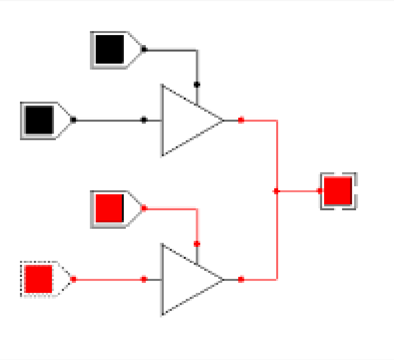
  - **Functie:**
    - Stuurlijn = 1 ➔ signaal wordt doorgegeven naar de uitgang
    - Stuurlijn = 0 ➔ signaal wordt niet doorgegeven
  - Tri-state buffers worden in paren gebruikt

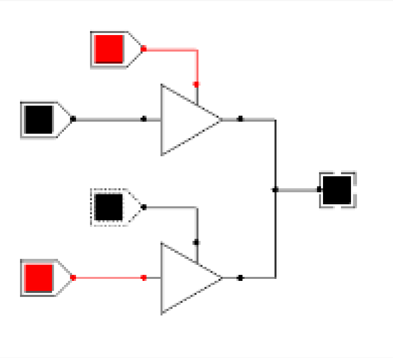
2.2. Basispoorten (vervolg)

- **Tri-state buffer :** verschillende toestanden die zich kunnen voordoen



Alle tri-state buffers ontkoppeld:
⇒ = stuurlijnen zijn 0
⇒ geen signaal op de geleider
⇒ uitgang is onbepaald (-/)

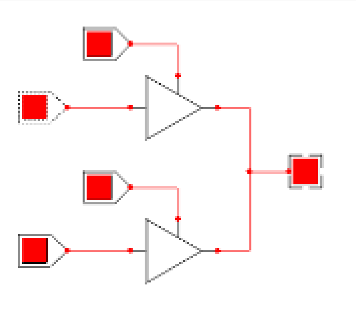




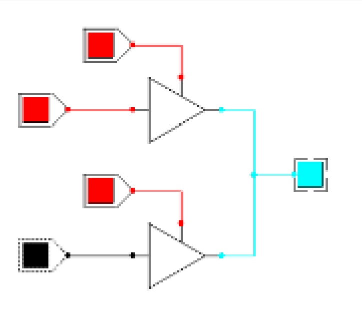
Juist één buffer gekoppeld
⇒ = juist één buffer met stuurlijn 1
⇒ dan komt het signaal van die buffer op de geleider

2.2. Basispoorten (vervolg)

- **Tri-state buffer :** verschillende toestanden die zich kunnen voordoen



Meerdere buffers gekoppeld met het zelfde signaal
→ dat signaal op de geleider



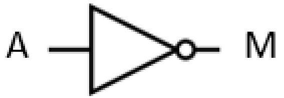
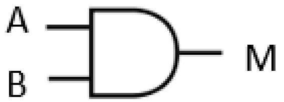
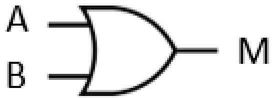
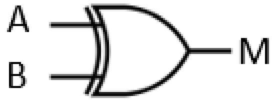
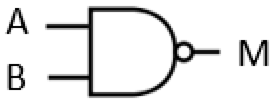
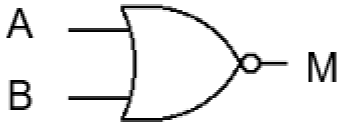
Meerdere buffers gekoppeld met verschillend signaal
→ **kortsluiting !!!** (x/)

3-state buffers worden net gebruikt om dit soort kortsluitingen te vermijden.

HO GENT

2.2. Basispoorten (vervolg)

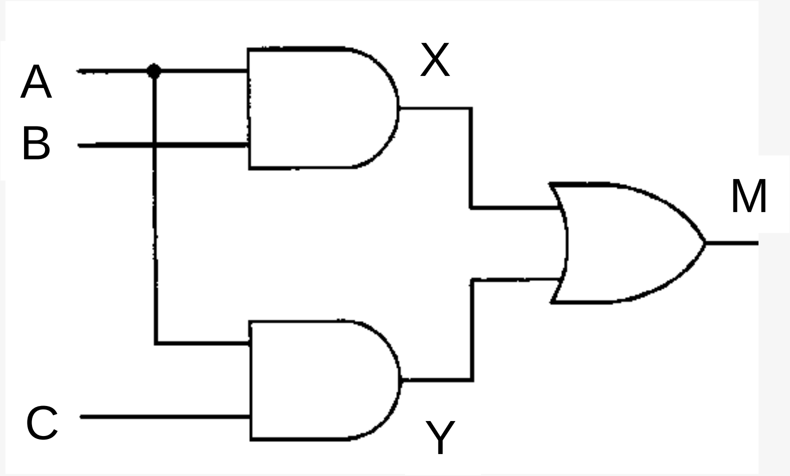
- **Overzicht**

<table>
<tr><td>NOT</td><td></td></tr>
<tr><td>AND</td><td></td></tr>
<tr><td>OR</td><td></td></tr>
<tr><td>XOR</td><td></td></tr>
<tr><td>NAND</td><td></td></tr>
<tr><td>NOR</td><td></td></tr>
</table>

2.3. Logische Schakelingen

Voor complexere netwerken:

- werk vanuit de ingangen naar de uitgangen toe
- noteer tussenresultaten, zoveel als nodig

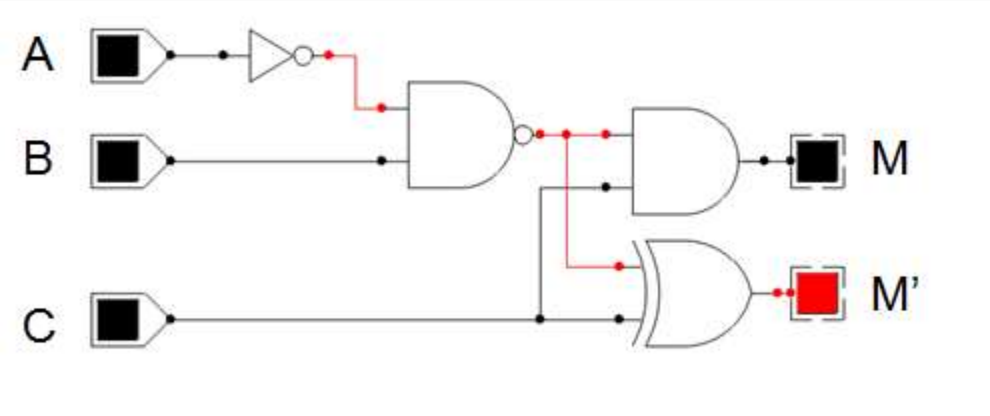


A	B	C	X	Y	M
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

HO
GENT

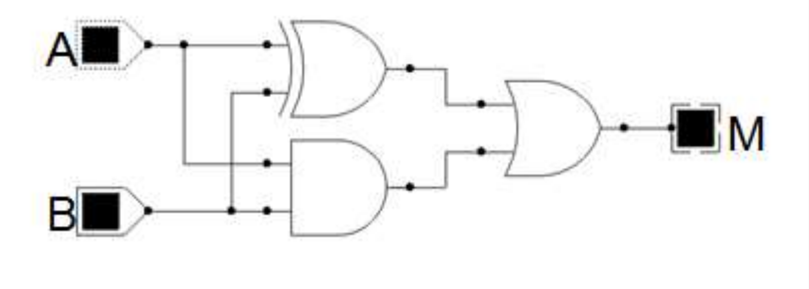
Oefeningen op logische schakelingen

- **Oefening 1:** stel de waarheidstabel op voor volgende schakeling



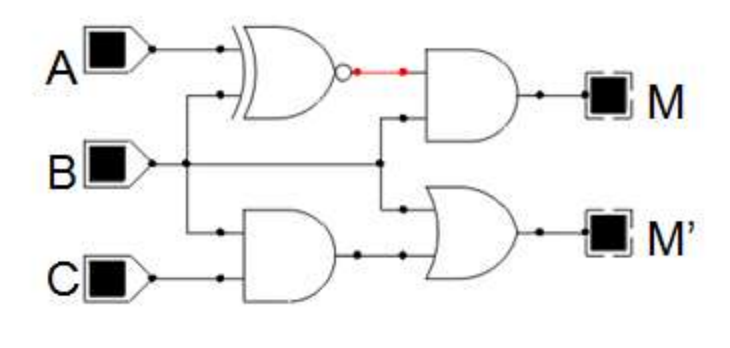
Oefeningen op logische schakelingen

- **Oefening 2:** stel de waarheidstabel op voor volgende schakeling



Oefeningen op logische schakelingen

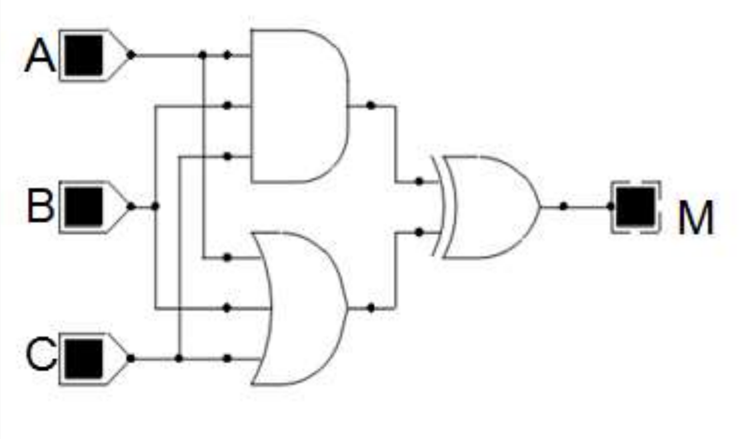
- **Oefening 3:** stel de waarheidstabel op voor volgende schakeling



HO GENT

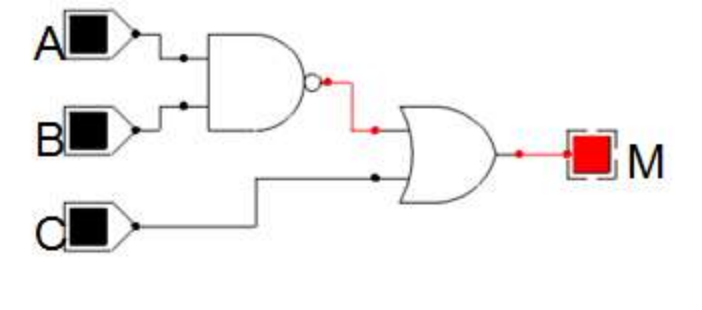
Oefeningen op logische schakelingen

- **Oefening 4:** stel de waarheidstabel op voor volgende schakeling



Oefeningen op logische schakelingen

- **Oefening 5:** stel de waarheidstabel op voor volgende schakeling

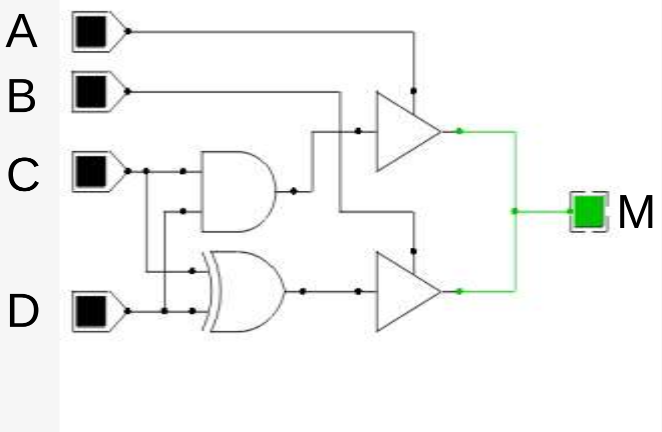


Oefeningen op logische schakelingen

- **Oefening 6:** stel de waarheidstabel op voor volgende schakeling. Gebruik volgende symbolen voor speciale situaties:

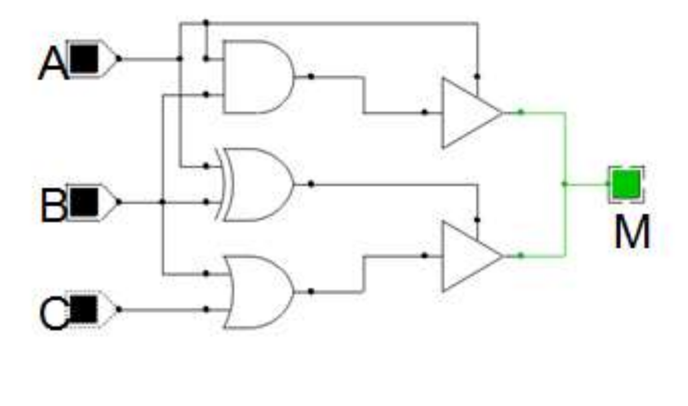
  x = kortsluiting

  - = onbepaald



Oefeningen op logische schakelingen

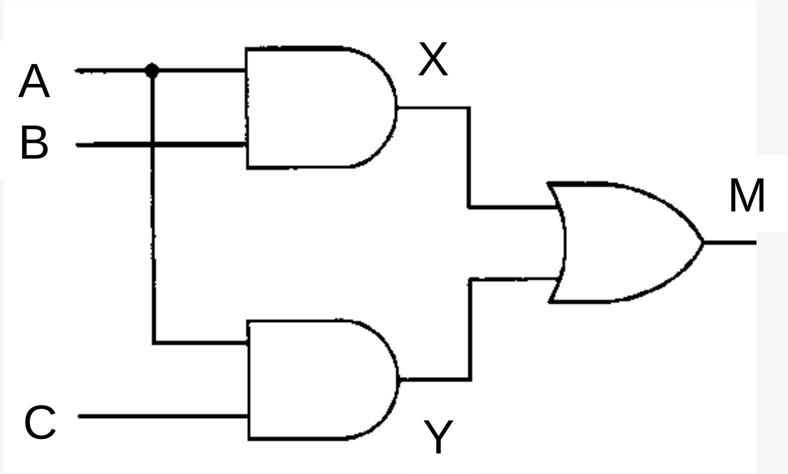
- **Oefening 7:** stel de waarheidstabel op voor volgende schakeling

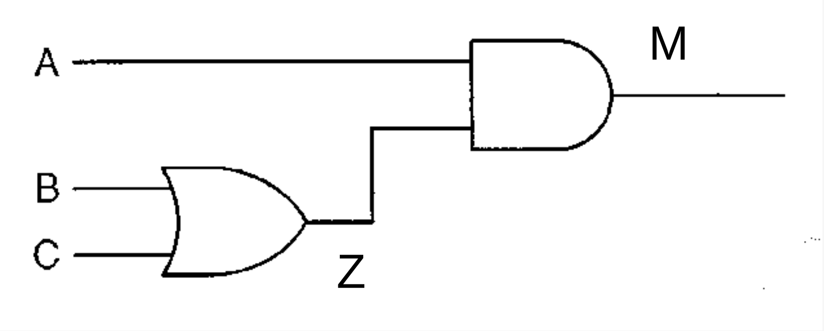


HO
GENT

2.4. Equivalente schakelingen

- We noemen twee logische schakelingen equivalent wanneer ze **dezelfde waarheidstabel** hebben. Bijvoorbeeld:
  - Werk zelf de 2 waarheidstabellen uit!





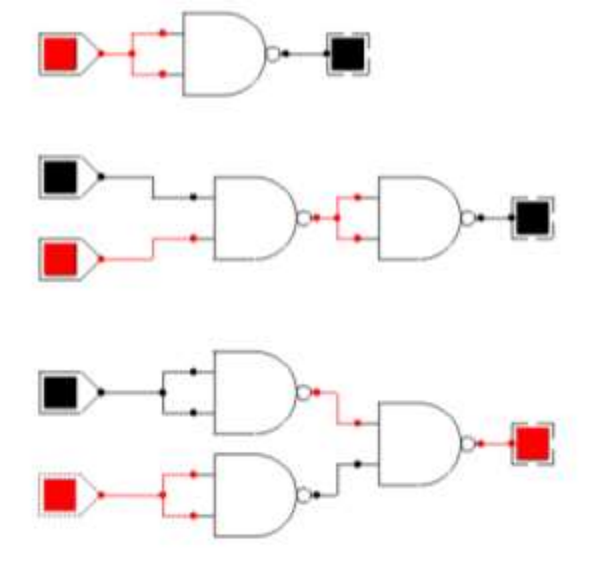
2.4. Equivalente schakelingen (vervolg)

- Equivalentie bij basispoorten:
  - In sommige technologieën is het alleen mogelijk om NAND- of NOR- poorten te maken met transistoren.
  - **Oplossing:** de andere basispoorten bouwen als schakelingen van NAND- of NOR-poorten. Bijvoorbeeld:

NIET-basispoort:

EN-basispoort:

OF-basispoort:



HO GENT

IT Fundamentals hoofdstuk 3

Boole-algebra's

HO
GENT

Inhoud.

HO GENT

3.1. Definitie

Een Boole-algebra is één specifieke algebraïsche structuur.

Een Boole-algebra **B** bestaat uit

- een verzameling S die **minstens twee constanten**, **0** en **1**, bevat;
- twee **binaire operatoren** op S: **+** en **·** ;
- een **unaire operator** op S: **–** (complement).

HO GENT

3.1. Definitie (vervolg)

In een Boole-algebra moeten voor alle x, y, z ∈ S **de axioma's van Huntington** geldig zijn:

1. Commutatieve **wetten**:

$$x + y = y + x$$
$$x \cdot y = y \cdot x$$

2. Distributieve **wetten**:

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$$
$$x + (y \cdot z) = (x + y) \cdot (x + z)$$

3. Identiteit**swetten**:

$$x + 0 = x$$
$$x \cdot 1 = x$$

4. Complement**wetten**:

$$x+\overline{x}=1$$
$$x . \overline{x}= 0$$

Notatie: $B = (S, +, \cdot, -, 0, 1) = (S, +, \cdot, ^{-})$

Let op! De wiskundige symbolen +, ·, 0 en 1 die hier gebruikt worden voor het definiëren van een Boole-algebra hebben NIET dezelfde betekenis als de +, ·, 0 en 1 die gebruikt worden binnen bv. de theorie van de reële getallen.

HO GENT

3.1. Definitie (vervolg)

→ **Praktische afspraken:**

- De binaire operator **. moet niet steeds expliciet opgeschreven worden**, zo is $x \cdot y = xy$
- Bij de uitwerking van een uitdrukking heeft de bewerking **· steeds voorrang op +** tenzij haakjes een andere volgorde aangeven

HO GENT

3.1. Definitie (vervolg)

3.1.1 Voorbeeld:

Stel B = {0, 1}.
Op deze verzameling B worden drie operatoren gedefinieerd:

- twee binaire operatoren, nl. **+** en **·**
- één unaire operator, **–**

De definitie van deze drie operatoren wordt gegeven a.d.h.v. de volgende tabellen:

+	0	1
0	0	1
1	1	1

·	0	1
0	0	0
1	0	1

	−
0	1
1	0

Is de structuur (B, +, ·, − , 0, 1) een Boole-algebra?

HO
GENT

3.1. Definitie (vervolg)

3.1.1 Voorbeeld (vervolg):

Verificatie van axioma’s van Huntington:

Stel a, b, c ∈ B dan geldt:

1. **Commutativiteit:** $a + b = b + a$

   $a \cdot b = b \cdot a$

OK

Bewijs:

a	b	$a+b$	$b+a$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	1

Als de matrix diagonaal symmetrisch is, is de operator commutatief

+	0	1
0	0	1
1	1	1

Analoog voor ·

3.1. Definitie (vervolg)

3.1.1 Voorbeeld (vervolg):

2. Distributiviteit: $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

$a + (b \cdot c) = (a + b) \cdot (a + c)$

OK

Bewijs:

a	b	c	$b \cdot c$	$a + (b \cdot c)$	$a + b$	$a + c$	$(a + b) \cdot (a + c)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

HO GENT

3.1. Definitie (vervolg)

3.1.2 Voorbeeld (vervolg):

3. Identiteit: $a + 0 = a$

$a \cdot 1 = a$

Bewijs:

+	0	1
0	0	1
1	1	1

·	0	1
0	0	0
1	0	1

OK

4. Complement: $a + \bar{a} = 1$

$a \cdot \bar{a} = 0$

Bewijs:

a	$\bar{a}$	$a + \bar{a}$	$a \cdot \bar{a}$
0	1	1	0
1	0	1	0

OK

3.1. Definitie (vervolg)

1. **Commutatieve wetten**:

   $x + y = y + x$

   $x \cdot y = y \cdot x$

   OK

2. **Distributieve wetten**:

   $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$

   $x + (y \cdot z) = (x + y) \cdot (x + z)$

   OK

3. **Identiteitswetten**:

   $x + 0 = x$

   $x \cdot 1 = x$

   OK

4. **Complementwetten**:

   $x+\overline{x}=1$

   $x \,.\, \overline{x}= 0$

   OK

Alle 4 axioma's voldaan → Dit is een Boole algebra!

HO GENT

3.1. Definitie (vervolg)

3.1.3 Oefeningen:

1. Noem D de verzameling gehele delers van 6.

   Definieer de volgende operatoren op D:

$$\begin{aligned} x + y &= \mathrm{kgv}(x, y) \\ x \cdot y &= \mathrm{ggd}(x, y) \\ \overline{x} &= 6/x. \end{aligned}$$

Is (D, +, ·, −) een Boole-algebra? Motiveer uw antwoord.

2. Noem D de verzameling gehele delers van 8.

Definieer de volgende operatoren op D:

$$\begin{aligned} x + y &= \mathrm{kgv}(x, y) \\ x \cdot y &= \mathrm{ggd}(x, y) \\ \overline{x} &= 8/x. \end{aligned}$$

Is (D, +, ·, −) een Boole-algebra? Motiveer uw antwoord.

3.2. Dualiteitsbeginsel

Duale uitdrukking: Alle symbolen vervangen door duale vorm

symbool	duale vorm
$+$	$\cdot$
$\cdot$	$+$
0	1
1	0
$-$	$-$

Uitdrukking is geldig ⇔ Duale is geldig

HO GENT

3.2. Dualiteitsbeginsel

axioma	duale vorm
$a + b = b + a$	$a \cdot b = b \cdot a$
$a \cdot (b + c) = (a \cdot b) + (a \cdot c)$	$a + (b \cdot c) = (a + b) \cdot (a + c)$
$a + 0 = a$	$a \cdot 1 = a$
$a + \overline{a} = 1$	$a \cdot \overline{a} = 0$

3.3. Eigenschappen

3.3.1 Eigenschap 1: het complement is uniek

Voor elke x ∈ S bestaat er juist één x ∈ S zodat $x + \bar{x} = 1 \; en \; x \cdot \bar{x} = 0$

Bewijs: Stel dat er nog een element y ∈ S bestaat waarvoor x + y = 1 en x · y = 0.

We tonen aan dat $y = \bar{x}$

$$
\begin{aligned}
y &= y \cdot 1 && \text{(identiteitswet)} \\
&= y \cdot (x + \overline{x}) && \text{(complementwet)} \\
&= y \cdot x + y \cdot \overline{x} && \text{(distributieve wet)} \\
&= x \cdot y + \overline{x} \cdot y && \text{(commutatieve wet)} \\
&= 0 + \overline{x} \cdot y && \text{(veronderstelling)} \\
&= x \cdot \overline{x} + \overline{x} \cdot y && \text{(complementwet)} \\
&= \overline{x} \cdot x + \overline{x} \cdot y && \text{(commutatieve wet)} \\
&= \overline{x} \cdot (x + y) && \text{(distributieve wet)} \\
&= \overline{x} \cdot 1 && \text{(veronderstelling)} \\
&= \overline{x} && \text{(identiteitswet)}
\end{aligned}
$$

HO GENT

3.3. Eigenschappen (vervolg)

3.3.2 Eigenschap 2: involutie

Als we de unaire operator tweemaal toepassen op een element x dan krijgen we als uitkomst opnieuw x

$$\bar{\bar{x}} = x$$

Bewijs:

$$\begin{aligned}
\bar{\bar{x}} &= \bar{\bar{x}} \cdot 1 && \text{(identiteitswet)} \\
&= \bar{\bar{x}} \cdot (x + \bar{x}) && \text{(complementwet)} \\
&= (\bar{\bar{x}} \cdot x) + (\bar{\bar{x}} \cdot \bar{x}) && \text{(distributieve wet)} \\
&= (\bar{\bar{x}} \cdot x) + (\bar{x} \cdot \bar{\bar{x}}) && \text{(commutatieve wet)} \\
&= (\bar{\bar{x}} \cdot x) + 0 && \text{(complementwet)} \\
&= (\bar{\bar{x}} \cdot x) + (x \cdot \bar{x}) && \text{(complementwet)} \\
&= (x \cdot \bar{\bar{x}}) + (x \cdot \bar{x}) && \text{(commutatieve wet)} \\
&= x \cdot (\bar{\bar{x}} + \bar{x}) && \text{(distributieve wet)} \\
&= x \cdot (\bar{x} + \bar{\bar{x}}) && \text{(commutatieve wet)} \\
&= x \cdot 1 && \text{(complementwet)} \\
&= x && \text{(identiteitswet)}
\end{aligned}$$

3.3. Eigenschappen (vervolg)

3.3.3 Eigenschap 3: complement van 0 en 1

Het complement van 0 is 1 en omgekeerd.

$$\begin{aligned}\overline{0} &= 1\\ \overline{1} &= 0\end{aligned}$$

Opmerking: Beide uitdrukking zijn elkaars duale.

$$\begin{aligned}\overline{0} &= \overline{0}+0 && \text{(identiteitswet)}\\ &= 1 && \text{(complementwet)}\end{aligned} \qquad \begin{aligned}\overline{1} &= \overline{1}\cdot 1\\ &= 0\end{aligned}$$

Beide bewijzen zijn ook elkaars duale!

3.3. Eigenschappen (vervolg)

3.3.4 Eigenschap 4: idempotentie

Wanneer de operator + toegepast wordt op tweemaal hetzelfde argument dan is het resultaat dat argument.

Wanneer de operator · toegepast wordt op tweemaal hetzelfde argument dan is het resultaat dat argument.

$$\begin{aligned} x + x &= x \\ x \cdot x &= x \end{aligned}$$

3.3.5 Eigenschap 5: begrenzing

Om het even welke waarde optellen bij 1 levert steeds 1. Om het even welke waarde vermenigvuldigen met 0 levert steeds 0.

$$\begin{aligned} x + 1 &= 1 \\ x \cdot 0 &= 0 \end{aligned}$$

3.3. Eigenschappen (vervolg)

3.3.6 Eigenschap 6: absorptie

Een element x optellen bij een product waarin x voorkomt als factor levert steeds x.

Een element x vermenigvuldigen met een som waarin x als term voorkomt levert steeds

$$\begin{aligned} x + (x \cdot y) &= x \\ x \cdot (x + y) &= x \end{aligned}$$

3.3.7 Eigenschap 7: associatief

Wanneer we drie elementen vermenigvuldigen dan kunnen we de haakjes verplaatsen zonder het resultaat te wijzigen. We kunnen de haakjes dus ook weglaten omdat de uitdrukking niet verkeerd kan begrepen worden.

$$\begin{aligned} x + (y + z) &= (x + y) + z = x + y + z \\ x \cdot (y \cdot z) &= (x \cdot y) \cdot z = x \cdot y \cdot z \end{aligned}$$

HO GENT

3.3. Eigenschappen (vervolg)

3.3.8 Eigenschap 8: wetten van de Morgan

Het complement van een som is het product van de complementen.

Het complement van een product is de som van de complementen.

$$\overline{x+y} = \overline{x} \cdot \overline{y}$$
$$\overline{x \cdot y} = \overline{x} + \overline{y}$$

3.4. Oefeningen

1. Geef het bewijs van de vierde, vijfde en zesde eigenschap: idempotentie, begrenzing en absorptie.
2. Hoe luiden de eigenschappen 5 en 6 voor de Boole-algebra (P(U), ∪, ∩, − , ∅, U).
3. Stel dat (S, +, ·, − , 0, 1) een Boole-algebra is met x, y, z ∈ S. Bepaal het complement en de duale uitdrukking van:

   a) $x + \overline{y \cdot z}$

   b) $(x \cdot y \cdot z) + (\overline{x} \cdot \overline{y} \cdot \overline{z})$.

4. Stel dat (S, +, ·, − , 0, 1) een Boole-algebra is met x, y, z, u ∈ S. Toon de geldigheid van de volgende uitdrukkingen aan door te steunen op de axioma's en eigenschappen van een Boole-algebra.

   a) $(x + \overline{y}) \cdot (\overline{x} + y) = \overline{x} \cdot \overline{y} + x \cdot y$

   b) $x \cdot y + \overline{y} \cdot z + x \cdot z = x \cdot y + \overline{y} \cdot z$

   c) $x \cdot \overline{z} + \overline{x} \cdot \overline{z} + \overline{y} \cdot z = \overline{y} + \overline{z}$

   d) $x \cdot y + \overline{x \cdot \overline{y} + \overline{x} \cdot \overline{y}} = y$

   e) $(x \cdot y + x \cdot u) \cdot (z \cdot u + \overline{x} \cdot u) \cdot (\overline{y} \cdot z + \overline{x} \cdot \overline{y}) \cdot x \cdot \overline{y} = x \cdot \overline{y} \cdot z \cdot u$

IT Fundamentals hoofdstuk 4

Boolese uitdrukkingen en functies

HO
GENT

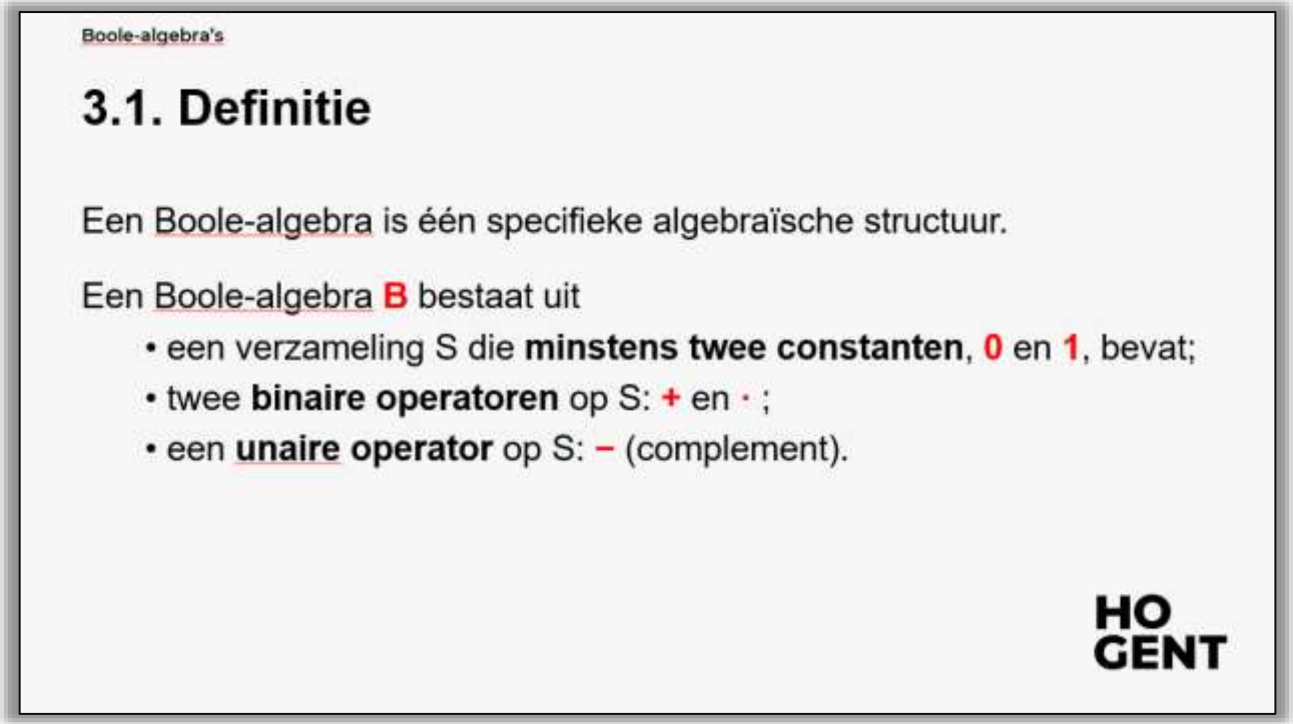
Inhoud.

HO
GENT

4.1. Boolese uitdrukkingen

We beperken ons tot de **eenvoudigste** Boole-algebra, de **minimale Boole-algebra** B.

Herhaling:



Boole-algebra's

3.1. Definitie

Een Boole-algebra is één specifieke algebraïsche structuur.

Een Boole-algebra **B** bestaat uit

- een verzameling S die **minstens twee constanten**, **0** en **1**, bevat;
- twee **binaire operatoren** op S: **+** en · ;
- een **unaire operator** op S: **–** (complement).

4.1. Boolese uitdrukkingen

Minimale Boole-algebra:

- een verzameling S die ~~minstens~~ **twee constanten**, **0** en **1**, bevat;

$$S = \{0,1\}$$

- twee **binaire operatoren** op S: **+** en **·** ;

+	0	1
0	0	1
1	1	1

·	0	1
0	0	0
1	0	1

- een **unaire operator** op S: **‒** (complement).

$$\overline{1} = 0$$

$$\overline{0} = 1$$

HO GENT

4.1. Boolese uitdrukkingen

Definitie 1: Een Boolese functie is van de vorm $f: B^n \rightarrow B: (x_1, x_2, \dots, x_n) \mapsto f(x_1, x_2, \dots, x_n)$

met $f(x_1, x_2, \dots, x_n)$ een Boolese uitdrukking.

Voorstelling:

Uitdrukking

$$f(x, y, z) = (x + y) \cdot z$$

Waarheidstabel

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

HO GENT

4.1. Boolese uitdrukkingen (vervolg)

Voorbeeld 1: $f : B^2 \to B : (x,y) \mapsto f(x,y) = x + \overline{y} + \overline{x} \cdot y.$

De corresponderende tabel is

x	y	$\overline{x}$	$\overline{y}$	$\overline{x} \cdot y$	$f(x,y)$
0	0	1	1	0	1
0	1	1	0	1	1
1	0	0	1	0	1
1	1	0	0	0	1

HO GENT

4.1. Boolese uitdrukkingen (vervolg)

Voorbeeld 2:

Beschouw de functie $f : B^3 \to B : (x, y, z) \mapsto x \cdot (\overline{y} + z)$

In dit geval geldt $f(x, y, z) = x \cdot (\overline{y} + z)$

De corresponderende tabel

x	y	z	$\overline{y}$	$\overline{y} + z$	$x \cdot (\overline{y} + z)$
0	0	0	1	1	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	0	1	1

4.1. Boolese uitdrukkingen (vervolg)

Twee Boolese functies zijn gelijk als ze voor dezelfde input steeds dezelfde output hebben.

(Officieel: Twee Boolese functies, f en g, in n veranderlijken zijn gelijk als en slechts als

$$f(x_1, x_2, \ldots, x_n) = g(x_1, x_2, \ldots, x_n)$$

voor alle $(x_1, x_2, \ldots, x_n) \in B^n$.)

Voorbeeld:

$$f(x,y) = (x+\overline{y}) \cdot \overline{x}$$
$$g(x,y) = \overline{x+y}.$$

x	y	$\overline{x}$	$\overline{y}$	$x+\overline{y}$	$x+y$	$f(x,y)$	$g(x,y)$
0	0	1	1	1	0	1	1
0	1	1	0	0	1	0	0
1	0	0	1	1	1	0	0
1	1	0	0	1	1	0	0

HO GENT

4.2. Minimale en maximale Boolese uitdrukkingen

Definitie: Een Boolese uitdrukking in n variabelen, x_1 , x_2 , . . . , x_n is **minimaal** als ze het **product** is van n factoren en waarbij de k-de factor x_k *of* $\overline{x}_k$ is.

Definitie: Een Boolese uitdrukking in n variabelen, x_1 , x_2 , . . . , x_n is **maximaal** als ze de **som** is van n termen en waarbij de k-de term x_k *of* $\overline{x}_k$ is.

Voorbeelden:

Minimale uitdrukkingen:

in één variabele x	in twee variabelen x en y	in drie variabelen x, y en z
x	$x \cdot y$	$x \cdot y \cdot z$
		$x \cdot y \cdot \overline{z}$
	$x \cdot \overline{y}$	$x \cdot \overline{y} \cdot z$
		$x \cdot \overline{y} \cdot \overline{z}$
$\overline{x}$	$\overline{x} \cdot y$	$\overline{x} \cdot y \cdot z$
		$\overline{x} \cdot y \cdot \overline{z}$
	$\overline{x} \cdot \overline{y}$	$\overline{x} \cdot \overline{y} \cdot z$
		$\overline{x} \cdot \overline{y} \cdot \overline{z}$

Maximale uitdrukkingen:

in één variabele x	in twee variabelen x en y	in drie variabelen x, y en z
x	$x + y$	$x + y + z$
		$x + y + \overline{z}$
	$x + \overline{y}$	$x + \overline{y} + z$
		$x + \overline{y} + \overline{z}$
$\overline{x}$	$\overline{x} + y$	$\overline{x} + y + z$
		$\overline{x} + y + \overline{z}$
	$\overline{x} + \overline{y}$	$\overline{x} + \overline{y} + z$
		$\overline{x} + \overline{y} + \overline{z}$

4.2. Minimale en maximale Boolese uitdrukkingen

Boolese functie kan in **2 standaardvormen** gezet worden

Gelijke functie = gelijke standaardvormen

- **disjunctieve normaalvorm** (**DNV**)
- **conjunctieve normaalvorm** (**CNV**)

Beide **uniek** voor elke functie

DNV = **som** van **minimale** uitdrukkingen

bv: $(x \cdot \bar{y} \cdot z) + (\bar{x} \cdot y \cdot \bar{z}) + (x \cdot y \cdot \bar{z})$

CNV = **product** van **maximale** uitdrukkingen

bv: $(x + \bar{y} + z) \cdot (\bar{x} + y + \bar{z}) \cdot (x + y + \bar{z})$

HO
GENT

4.3. De disjunctieve normaalvorm: DNV

2 methoden om uitdrukkingen om te zetten in DNV

1. Uitdrukking **herleiden** mbv axioma's en eigenschappen
2. Aflezen uit **waarheidstabel**

4.3. De disjunctieve normaalvorm: DNV (vervolg)

4.3.1. 1ste Methode: gebruik axioma's en eigenschappen

Voorbeeld 1:

Stel $f : B^3 \to B : x \mapsto f(x,y,z) = y\bar{z} + (y+z)(\overline{z+\bar{x}})$.

$$\begin{array}{rcll}
f(x,y,z) & = & y\bar{z} + (y+z)(\overline{z+\bar{x}}) & \\
& = & y\bar{z} + (y+z)\bar{z}\,\bar{\bar{x}} & \text{(wet van de Morgan)} \\
& = & y\bar{z} + (y+z)\bar{z}x & \text{(involutie)} \\
& = & y\bar{z} + y\bar{z}x + z\bar{z}x & \text{(distributieve wet)} \\
& = & y\bar{z} + y\bar{z}x + 0x & \text{(complementwet)} \\
& = & y\bar{z} + y\bar{z}x & \text{(begrenzing, identiteitswet)} \\
& = & 1y\bar{z} + y\bar{z}x & \text{(identiteitswet)} \\
& = & (x+\bar{x})y\bar{z} + y\bar{z}x & \text{(complementwet)} \\
& = & xy\bar{z} + \bar{x}y\bar{z} + y\bar{z}x & \text{(distributieve wet)} \\
& = & xy\bar{z} + \bar{x}y\bar{z} & \text{(idempotentie, commutatieve wet)}
\end{array}$$

Besluit: $f(x,y,z) = (x \cdot y \cdot \bar{z}) + (\bar{x} \cdot y \cdot \bar{z})$ (DNV).

HO GENT

4.3. De disjunctieve normaalvorm: DNV (vervolg)

4.3.1. 1ste Methode: gebruik axioma's en eigenschappen (vervolg)

Voorbeeld 2:

Stel $f : B^3 \to B : x \mapsto f(x,y,z) = (x \cdot \overline{y} + \overline{x} \cdot y) \cdot (x + z)$.

$$
\begin{aligned}
& f(x,y,z) \\
&= (x \cdot \overline{y} + \overline{x} \cdot y) \cdot (x + z) \\
&= x \cdot \overline{y} \cdot x + \overline{x} \cdot y \cdot x + x \cdot \overline{y} \cdot z + \overline{x} \cdot y \cdot z && \text{(distributieve wet)} \\
&= x \cdot \overline{y} + 0 \cdot y + x \cdot \overline{y} \cdot z + \overline{x} \cdot y \cdot z && \text{(comm. wet, idempotentie, compl. wet)} \\
&= x \cdot \overline{y} + x \cdot \overline{y} \cdot z + \overline{x} \cdot y \cdot z && \text{(begrenzing, identiteitswet)} \\
&= x \cdot \overline{y} + \overline{x} \cdot y \cdot z && \text{(absorptie)} \\
&= x \cdot \overline{y} \cdot 1 + \overline{x} \cdot y \cdot z && \text{(identiteitswet)} \\
&= x \cdot \overline{y} \cdot (z + \overline{z}) + \overline{x} \cdot y \cdot z && \text{(complementwet)} \\
&= x \cdot \overline{y} \cdot z + x \cdot \overline{y} \cdot \overline{z} + \overline{x} \cdot y \cdot z && \text{(distributieve wet)}
\end{aligned}
$$

Besluit: $f(x,y,z) = (x \cdot \overline{y} \cdot z) + (x \cdot \overline{y} \cdot \overline{z}) + (\overline{x} \cdot y \cdot z)$ (DNV).

4.3. De disjunctieve normaalvorm: DNV (vervolg)

4.3.1. 2de Methode: gebruik de uitvoertabel

In het lang uitgeschreven:

- Elke rij heeft een **minimale term**.

Deze bevat x voor de rijen waar $x = 1$, en $\bar{x}$ waar $x = 0$. Analoog voor y, z, …

- **Vermenigvuldig** deze minimale term met de **output** voor elke rij

x	y	$f(x,y) = x + \bar{x}y$
0	0	0
0	1	1
1	0	1
1	1	1

$$f(0{,}0) \cdot (\bar{x} \cdot \bar{y}) = 0 \cdot (\bar{x} \cdot \bar{y})$$
$$f(0{,}1) \cdot (\bar{x} \cdot y) = 1 \cdot (\bar{x} \cdot y)$$
$$f(1{,}0) \cdot (x \cdot \bar{y}) = 1 \cdot (x \cdot \bar{y})$$
$$f(1{,}1) \cdot (x \cdot y) = 1 \cdot (x \cdot y)$$

- **Tel** deze **op:** $f(x,y) = 0 \cdot (\bar{x} \cdot \bar{y}) + 1 \cdot (\bar{x} \cdot y) + 1 \cdot (x \cdot \bar{y}) + 1 \cdot (x \cdot y)$

- Gebruik identiteit en begrenzing om de 0 en 1 weg te werken:

$$f(x,y) = (\bar{x} \cdot y) + (x \cdot \bar{y}) + (x \cdot y)$$

4.3. De disjunctieve normaalvorm: DNV (vervolg)

4.3.1. **2de Methode: gebruik de uitvoertabel**

Conclusie:

- We hebben enkel de rijen nodig waar de uitvoer 1 is

$$f(x, y) = (\bar{x} \cdot y) + (x \cdot \bar{y}) + (x \cdot y)$$

x	y	$f(x,y) = x + \bar{x}y$
0	0	0
$\boxed{0}$	$\boxed{1}$	$\boxed{1}$
$\boxed{1}$	$\boxed{0}$	$\boxed{1}$
$\boxed{1}$	$\boxed{1}$	$\boxed{1}$

4.3. De disjunctieve normaalvorm: DNV (vervolg)

4.3.1. 2de Methode: gebruik de uitvoertabel (vervolg)

Voorbeeld 2:

Stel $f : B^3 \rightarrow B : x \mapsto x \cdot (\overline{y} + z)$

x	y	z	$\overline{y}$	$\overline{y}+z$	$f(x,y,z)$
0	0	0	1	1	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	0	1	1

DNV:

$$f(x,y,z) = (x \cdot y \cdot z) + (x \cdot \overline{y} \cdot z) + (x \cdot \overline{y} \cdot \overline{z})$$

HO GENT

4.4. De conjunctieve normaalvorm: CNV

Gelijkenissen met DNV:

- Elke uitdrukking heeft 1 CNV
- Gelijke uitdrukkingen hebben gelijke CNV's
- 2 methodes om de CNV te bepalen: met eigenschappen en via waarheidstabel

Verschillen:

- CNV is een **product** van **maximale** uitdrukkingen (DNV: som van minimale)

Voorbeelden:

- $f(x,y) = (x+y)\cdot(\overline{x}+y)$
- $f(x,y,z) = (\overline{x}+y+z)\cdot(\overline{x}+\overline{y}+z)\cdot(\overline{x}+\overline{y}+\overline{z})$.

4.4. De conjunctieve normaalvorm: CNV (vervolg)

Methode: gebruik de uitvoertabel

De oplossingsmethode voor het bepalen van de CNV komt overeen met de **duale werkwijze**: dezelfde stappen worden doorlopen maar telkens de duale vorm ervan.

Methode DNV	Methode CNV
Stel de outputtabel op.	Stel de outputtabel op.
Zoek alle outputwaarden gelijk aan **1**.	Zoek alle outputwaarden gelijk aan **0**.
Construeer de bijhorende **minimale term**: • als input $x_i = \mathbf{1}$ dan x_i in minimale term • als input $x_i = \mathbf{0}$ dan $\overline{x_i}$ in minimale term.	Construeer de bijhorende **maximale uitdrukking**: • als input $x_i = \mathbf{0}$ dan x_i in maximale uitdrukking • als input $x_i = \mathbf{1}$ dan $\overline{x_i}$ in maximale uitdrukking.
Verbind alle minimale termen met een $+$.	Verbind alle maximale uitdrukkingen met een $\cdot$.

HO
GENT

4.4. De conjunctieve normaalvorm: CNV (vervolg)

Voorbeeld:

Stel $f : B^3 \to B : x \mapsto x \cdot (\overline{y} + z)$

De outputtabel voor f is:

x	y	z	$\overline{y}$	$\overline{y} + z$	$f(x,y,z)$
0	0	0	1	1	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	1	0	1	0
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	0	1	1

CNV:

$$f(x,y,z) = (x + y + z) \cdot (x + y + \overline{z}) \cdot (x + \overline{y} + z) \cdot (x + \overline{y} + \overline{z}) \cdot (\overline{x} + \overline{y} + z).$$

HO GENT

4.4. Vereenvoudigen van Boolese uitdrukkingen

De DNV en CNV van een functie **zijn uniek**. **Gelijke functies** hebben **eenzelfde DNV** en **CNV.**

CNV en DNV kunnen soms **verder vereenvoudigd** worden: vb

$$f(x,y) = (x \cdot \bar{y}) + (x \cdot y)$$
$$\Leftrightarrow f(x,y) = x$$

Een Boolese functie vereenvoudigen kan o.a. door te steunen op de axioma's en eigenschappen van een Boole-algebra.

Welke axioma's en eigenschappen er moeten toegepast worden om de uitdrukking te vereenvoudigen, daar zijn geen vaste regels voor. Het beste resultaat vindt men via **trial and error**.

HO GENT

4.6. Oefeningen

Oefening 1:

Bepaal de DNV en CNV van f(x, y, z) als:

a) f(1, 1, 1) = f(1, 0, 1) = f(1, 1, 0) = f(0, 0, 0) = 1. Alle andere waarden v. f zijn 0.
b) f(0, 0, 0) = f(0, 1, 0) = f(0, 1, 1) = 1. Alle andere waarden van f zijn 0.

Oefening 2:

Bepaal de DNV van f(x, y, z, u) als:

a) f(1, 1, 1, 1) = f(1, 0, 0, 1) = f(1, 0, 1, 0) = 1. Alle andere waarden van f zijn 0.
b) f(1, 0, 1, 0) = f(0, 0, 0, 0) = f(0, 1, 0, 1) = f(1, 1, 1, 1) = 1. Alle andere waarden van f zijn 0.

4.6. Oefeningen

Oefening 1: oplossing

a) De outputtabel voor f is:

x	y	z	$f(x,y,z)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

- De DNV is: $f(x,y,z) = (\overline{x} \cdot \overline{y} \cdot \overline{z}) + (x \cdot \overline{y} \cdot z) + (x \cdot y \cdot \overline{z}) + (x \cdot y \cdot z)$
- De CNV is: $f(x,y,z) = (x + y + \overline{z}) \cdot (x + \overline{y} + z) \cdot (x + \overline{y} + \overline{z}) \cdot (\overline{x} + y + z)$

HO
GENT

4.6. Oefeningen

Oefening 1: oplossing (vervolg)

b) De outputtabel voor f is:

x	y	z	$f(x,y,z)$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

- De DNV is:
  $f(x,y,z) = (\overline{x} \cdot \overline{y} \cdot \overline{z}) + (\overline{x} \cdot y \cdot \overline{z}) + (\overline{x} \cdot y \cdot z)$
- De CNV is:
  $f(x,y,z) = (x + y + \overline{z}) \cdot (\overline{x} + y + z) \cdot (\overline{x} + y + \overline{z}) \cdot (\overline{x} + \overline{y} + z) \cdot (\overline{x} + \overline{y} + \overline{z})$

HO
GENT

4.6. Oefeningen

Oefening 2: oplossing

Hier is enkel de DNV gevraagd. Voor de DNV moeten we voor alle outputwaarden die gelijk zijn aan 1 de corresponderende input kennen. Dit is precies hetgeen gegeven is..

a) $f(x,y,z,u) = (x \cdot y \cdot z \cdot u) + (x \cdot \overline{y} \cdot \overline{z} \cdot u) + (x \cdot \overline{y} \cdot z \cdot \overline{u})$

b) $f(x,y,z,u) = (x \cdot \overline{y} \cdot z \cdot \overline{u}) + (\overline{x} \cdot \overline{y} \cdot \overline{z} \cdot \overline{u}) + (\overline{x} \cdot y \cdot \overline{z} \cdot u) + (x \cdot y \cdot z \cdot u)$

4.6. Oefeningen

Oefening 3:

Geef de DNV (op twee verschillende manieren) en CNV van:

a) $f(x,y,z) = \overline{x+y+z}$

b) $f(x,y,z) = ((\overline{x}+y)+\overline{z}) \cdot \overline{z} \cdot x$

c) $f(x,y,z) = (x \cdot (\overline{y}+z)) + \overline{z}$

HO GENT

4.6. Oefeningen

Oefening 3: oplossing

a) • De DNV:

- Eerste methode (mbv. eigenschappen en axioma's van een Boole-algebra):

$$\begin{aligned} f(x,y,z) &= \overline{x+y+z} && \text{(geg.)} \\ &= \overline{x}\cdot\overline{y}\cdot\overline{z} && \text{(wet van De Morgan)} \end{aligned}$$

- Tweede methode (via de outputtabel):

x	y	z	$x+y+z$	$f(x,y,z)=\overline{x+y+z}$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

Eén outputwaarde gelijk aan 1 levert één minimale term op voor de DNV: $f(x,y,z)=\overline{x}\cdot\overline{y}\cdot\overline{z}$

• De CNV (via de outputtabel):

Voor de CNV werken we met de outputwaarden 0 i.p.v. 1.

$f(x,y,z) = (x+y+\overline{z})\cdot(x+\overline{y}+z)\cdot(x+\overline{y}+\overline{z})\cdot(\overline{x}+y+z)\cdot(\overline{x}+y+\overline{z})\cdot(\overline{x}+\overline{y}+z)\cdot(\overline{x}+\overline{y}+\overline{z})$

HO GENT

4.6. Oefeningen

Oefening 3: oplossing

b) • De DNV:

– Eerste methode (mbv. eigenschappen en axioma's van een Boole-algebra):

$$\begin{aligned}
f(x,y,z) &= ((\overline{x}+y)+\overline{z})\cdot\overline{z}\cdot x && \text{(geg.)}\\
&= [(\overline{x}+y+\overline{z})\cdot\overline{z}]\cdot x && \text{(associatieve wet)}\\
&= \overline{z}\cdot x && \text{(absorptie)}\\
&= x\cdot\overline{z}\cdot 1 && \text{(commutatieve wet / identiteit)}\\
&= x\cdot\overline{z}\cdot(y+\overline{y}) && \text{(complementswet)}\\
&= (x\cdot y\cdot\overline{z})+(x\cdot\overline{y}\cdot\overline{z}) && \text{(distributieve wet / commutatieve wet)}
\end{aligned}$$

– Tweede methode (via de outputtabel):

x	y	z	$\overline{x}$	$\overline{z}$	$\overline{x}+y+\overline{z}$	$f(x,y,z)=(\overline{x}+y+\overline{z})\cdot\overline{z}\cdot x$
0	0	0	1	1	1	0
0	0	1	1	0	1	0
0	1	0	1	1	1	0
0	1	1	1	0	1	0
1	0	0	0	1	1	1
1	0	1	0	0	0	0
1	1	0	0	1	1	1
1	1	1	0	0	1	0

De DNV:

$f(x,y,z)=(x\cdot\overline{y}\cdot\overline{z})+(x\cdot y\cdot\overline{z})$

• De CNV (via de outputtabel):

$f(x,y,z)=(x+y+z)\cdot(x+y+\overline{z})\cdot(x+\overline{y}+z)\cdot(x+\overline{y}+\overline{z})\cdot(\overline{x}+y+\overline{z})$
$(\overline{x}+\overline{y}+\overline{z})$

HO
GENT

4.6. Oefeningen

Oefening 3: oplossing

c) • De DNV:

- Eerste methode (mbv. eigenschappen en axioma's van een Boole-algebra):

$$
\begin{aligned}
& f(x,y,z) \\
&= (x \cdot (\overline{y} + z)) + \overline{z} && \text{(geg)} \\
&= ((x \cdot \overline{y}) + (x \cdot z)) + \overline{z} && \text{(dist)} \\
&= (x \cdot \overline{y}) + (x \cdot z) + \overline{z} && \text{(assoc)} \\
&= (x \cdot \overline{y} \cdot 1) + (x \cdot z \cdot 1) + (\overline{z} \cdot 1 \cdot 1) && \text{(ident)} \\
&= (x \cdot \overline{y} \cdot (z + \overline{z})) + (x \cdot z \cdot (y + \overline{y})) + (\overline{z} \cdot (x + \overline{x}) \cdot (y + \overline{y})) && \text{(compl)} \\
&= (x \cdot \overline{y} \cdot z) + (x \cdot \overline{y} \cdot \overline{z}) + (x \cdot y \cdot z) + (x \cdot \overline{y} \cdot z) + \\
&\quad (x \cdot y \cdot \overline{z}) + (x \cdot \overline{y} \cdot \overline{z}) + (\overline{x} \cdot y \cdot \overline{z}) + (\overline{x} \cdot \overline{y} \cdot \overline{z}) && \text{(dist/comm)} \\
&= (x \cdot \overline{y} \cdot z) + (x \cdot \overline{y} \cdot \overline{z}) + (x \cdot y \cdot z) + (x \cdot y \cdot \overline{z}) + (\overline{x} \cdot y \cdot \overline{z}) + (\overline{x} \cdot \overline{y} \cdot \overline{z}) && \text{(idemp)}
\end{aligned}
$$

- Tweede methode (via de outputtabel):

x	y	z	$\overline{y}$	$\overline{y}+z$	$x \cdot (\overline{y}+z)$	$\overline{z}$	$f(x,y,z) = (x \cdot (\overline{y}+z)) + \overline{z}$
0	0	0	1	1	0	1	1
0	0	1	1	1	0	0	0
0	1	0	0	0	0	1	1
0	1	1	0	1	0	0	0
1	0	0	1	1	1	1	1
1	0	1	1	1	1	0	1
1	1	0	0	0	0	1	1
1	1	1	0	1	1	0	1

De DNV:
$f(x,y,z) = (\overline{x} \cdot \overline{y} \cdot \overline{z}) + (\overline{x} \cdot y \cdot \overline{z}) + (x \cdot \overline{y} \cdot \overline{z}) + (x \cdot \overline{y} \cdot z) + (x \cdot y \cdot \overline{z}) + (x \cdot y \cdot z)$

• De CNV (via de outputtabel):
$f(x,y,z) = (x + y + \overline{z}) \cdot (x + \overline{y} + \overline{z})$

HO GENT

4.6. Oefeningen

Oefening 4:

Vereenvoudig:

a) $(x + y) \cdot (\overline{y} \cdot \overline{z}) + (x \cdot \overline{y})$

b) $x \cdot \overline{y} + (\overline{x} + y) \cdot y$

c) $((x + y) + x \cdot z) \cdot (x + y \cdot z)$

d) $(x + x \cdot y + x \cdot y \cdot z) \cdot (x \cdot y + x \cdot z) \cdot (y + x \cdot y \cdot z)$

4.6. Oefeningen

Oefening 4: oplossing

Om deze vereenvoudigde vorm te vinden wordt de gegeven uitdrukking herleid d.m.v. de eigenschappen en axioma's van een Boole-algebra.

a)

$$
\begin{aligned}
(x+y)\cdot(\overline{y}\cdot\overline{z})+(x\cdot\overline{y}) &= (x\cdot\overline{y}\cdot\overline{z})+(y\cdot\overline{y}\cdot\overline{z})+(x\cdot\overline{y}) && \text{(distributieve wet)}\\
&= (x\cdot\overline{y}\cdot\overline{z})+(0\cdot\overline{z})+(x\cdot\overline{y}) && \text{(complementswet)}\\
&= (x\cdot\overline{y}\cdot\overline{z})+0+(x\cdot\overline{y}) && \text{(begrenzing)}\\
&= (x\cdot\overline{y}\cdot\overline{z})+(x\cdot\overline{y}) && \text{(identiteit)}\\
&= x\cdot\overline{y} && \text{(absorptie)}
\end{aligned}
$$

b)

$$
\begin{aligned}
x\cdot\overline{y}+(\overline{x}+y)\cdot y &= (x\cdot\overline{y})+y && \text{(absorptie)}\\
&= (x+y)\cdot(\overline{y}+y) && \text{(distributieve wet)}\\
&= (x+y)\cdot 1 && \text{(complementswet)}\\
&= x+y && \text{(identiteit)}
\end{aligned}
$$

HO GENT

4.6. Oefeningen

Oefening 4: oplossing (vervolg)

c)

$$
\begin{aligned}
((x+y)+x\cdot z)\cdot(x+y\cdot z) &= (x+y+x\cdot z)\cdot(x+y\cdot z) && \text{(associatieve wet)}\\
&= (x+y)\cdot(x+y\cdot z) && \text{(absorptie)}\\
&= x+(y\cdot y\cdot z) && \text{(distributieve wet)}\\
&= x+(y\cdot z) && \text{(idempotentie)}
\end{aligned}
$$

d)

$$
\begin{aligned}
&(x+x\cdot y+x\cdot y\cdot z)\cdot(x\cdot y+x\cdot z)\cdot(y+x\cdot y\cdot z)\\
&= (x+x\cdot y\cdot z)\cdot(x\cdot y+x\cdot z)\cdot y && \text{(absorptie)}\\
&= x\cdot(x\cdot y+x\cdot z)\cdot y && \text{(absorptie)}\\
&= (x\cdot y)\cdot((x\cdot y)+(x\cdot z)) && \text{(associatieve en commutatieve wet)}\\
&= x\cdot y && \text{(absorptie)}
\end{aligned}
$$

HO GENT